OPTIMAL CALL ADMISSION TO A MOBILE CELLULAR NETWORK

BY MOHAMMAD SAQUIB

A thesis submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Master of Science

Graduate Program in Electrical Engineering

Written under the direction of

Dr. Roy D. Yates

and approved by

_____

_____

_____

New Brunswick, New Jersey

October, 1994

ABSTRACT OF THE THESIS

Optimal Call Admission to a Mobile Cellular Network

by Mohammad Saquib, M.S.

Thesis Director: Dr. Roy D. Yates

To fulfill the future demand for and increase the reliability of mobile cellular service, we have explored the use of a call admission policy that constitutes a dynamic channel allocation scheme for a cellular system.

The call admission policy optimizes a weighted blocking criteria for a queueing network model of a mobile cellular system. In this context, the call admission problem is formulated as a Markov decision process (MDP). The value-iteration method has been applied to a uniformized chain of a small cellular system. The effectiveness of call admission policy for maximizing the call completion rate and for reducing handoff blocking of calls in progress is studied. In addition, several symmetry properties of the optimal policies which effectively truncate the size of policy space are mathematically established.

Table of Contents

Abstract ii

List of Figures vi

1. Introduction 1

1.1. Call Admission Policies 1

1.2. Single-Class and Two-Class Systems 2

1.3. Channel Allocation Schemes 2

1.4. Previous Work & Motivation 4

1.5. Model of the Cellular System 5

1.6. Outline of this Work 6

2. Markov Decision Processes (MDP) 7

2.1. Introduction to MDP 7

2.2. The Value-Iteration Algorithm for Discrete-Time Markov Decision Model 9

2.2.1. Value-Iteration Algorithm 10

2.2.2. Modified Value-Iteration Algorithm 11

2.3. Uniformized Continuous-Time Markov Chain 12

2.4. The Value-Iteration Algorithm for Continuous-Time Markov Decision Model 13

3. Markov Decision Process for a Cellular Network 15

3.1. Introduction 15

3.1.1. Chapter Outline 15

The System Parameters for an Arbitrary Linear Cellular System 15

MDP (Value-iteration algorithm) 18

3.1.2. Decomposed Modified Value-iteration Algorithm 20
3.2. Structural Properties of the Optimal Policy 21

4. Empirical Performance Measures for a Single-Class Cellular Network 23
4.1. Introduction 23
4.1.1. Definitions of the System Parameters 23
4.1.2. Performance Measures 25
Definition of the Parameters 25
Formulation of the Performance Measures 25
4.2. Maximizing Call Completion rate 27
4.2.1. Stationary Calls in the 4 Cell Ring System 27
4.2.2. Stationary Calls in the 3 Cell Linear System 28
4.2.3. Mobile Calls 30
4.2.4. Conclusion 32
4.3. Optimizing Weighted Reward rate 32
4.3.1. Performance Measures with Variable Weighting Factor 32
Purpose of the Weight 34
Selection of the Weight 34
4.3.2. Performance Measures with Different Handoff Rate 34
4.3.3. Performance Measures with Different Load 36

5. Empirical Performance Measures for a Two-Class Cellular Network 38
5.1. Introduction 38
5.1.1. Definitions of the Class-Wise System Parameters 39
5.1.2. Performance Measures 39
Definitions of the Parameters for Class-Wise Performance Measures 39
Class-Wise Performance Measures 39
Overall Performance Measures 39
5.2. Maximizing Call Completion Rate 41
5.2.1. Conclusion 43

5.3. Optimization of Weighted Call Incompletion Rate 43
5.4. Analysis of the Comparative Relation of Class-Wise Performance Measures 44
Slow and Fast Moving Customers 45
Short Class 1 Calls 47
5.4.1. Comparative Relation Of The Class-Wise Performance Measures 48

6. Conclusion and Future Work 57
6.1. Summary of Results 57
6.2. Suggestions for Further Study 58
References 59

List of Figures

2.1. Value-iteration algorithm for a discrete-time Markov chain. 10

2.2. Modified value-iteration algorithm for a discrete-time Markov chain. . . 11

2.3. Modified value-iteration algorithm for a continuous-time Markov chain. 14

3.1. Decomposed modified value-iteration algorithm for a continuous-time Markov chain. 20

4.1. Highway Linear Cell Model, Ba= base station, n = number of the cells. 24

4.2. City Ring Cell Model, Ba= base station, Co= conversation, Ha= handoff. 24

4.3. Average rate flow in a single-class mobile cellular system under the stationary policy is π. 26

4.4. Performance evaluation of a ring system by maximizing call completion rate under variable load with $c = 15$ and $c = 25$, when $n = 4$. 27

4.5. Performance evaluation of a linear system by maximizing the call completion rate under variable load with $n = 3$, $c = 15$. 28

4.6. Performance evaluation of a ring system by maximizing call completion rate under variable handoff rate with $n = 4$, $\lambda = 2.5$, $\mu = 1.0$, $c = 12$. . . 30

4.7. Performance evaluation of a ring system by maximizing call completion rate under variable handoff rate with $n = 4$, $\lambda = 7.5$, $\mu = 1.0$, $c = 12$. . . 31

4.8. Performance evaluation of a ring system by maximizing call completion rate under variable call arriving rate with $n = 4$, $\mu = 1.0$, $\alpha = 10.0$, $c = 10$. 31

4.9. Performance evaluation of a ring system by optimizing weighted call incompletion rate under variable weighting factor with $n = 4$, $\lambda = 1.0$, $\mu = 1.0$, $\alpha = 10.0$, $c = 10$. 33

4.10. Performance evaluation of a ring system by optimizing weighted call incompletion rate under variable weighting factor with $n = 4$, $\lambda = 1.25$, $\mu = 1.0$, $\alpha = 10.0$, $c = 10$. 33
4.11. Performance evaluation of a ring system by optimizing weighted call incompletion rate under variable handoff rate with $n = 4$, $\lambda = 1.0$, $\mu = 1.0$, $\omega = 10.0$, $c = 10$. 35
4.12. Performance evaluation of a ring system by optimizing weighted call incompletion rate under variable handoff rate with $n = 4$, $\lambda = 1.25$, $\mu = 1.0$, $\omega = 10.0$, $c = 10$. 35
4.13. Performance evaluation of a ring system by optimizing weighted call incompletion rate under variable new call arriving rate with $n = 4$, $\mu = 1.0$, $\alpha = 10.0$, $\omega = 5.0$, $c = 10$. 36
4.14. Performance evaluation of a ring system by optimizing weighted call incompletion rate under variable service rate with $n = 4$, $\lambda = 1.0$, $\alpha = 10.0$, $\omega = 5.0$, $c = 10$. 37
5.1. Class-wise average rate flow in a two-class mobile cellular system under the stationary policy π. 40
5.2. Performance evaluation of a two-class ring system by maximizing call completion rate under variable handoff rate of the class 2 calls with $n = 4$, $\lambda_1 = \lambda_2 = 0.2$, $\mu_1 = \mu_2 = 1.0$, $\alpha_1 = 0.0$, $c = 6$. 41
5.3. Performance evaluation of a two-class ring system by maximizing call completion rate under variable service rate of class 1 calls with $n = 4$, $\lambda_1 = \lambda_2 = 0.2$, $\mu_2 = 1.0$, $\alpha_1 = 1.0$, $\alpha_2 = 5.0$, $c = 6$. 42
5.4. Experimental class-wise system parameters and performance measures for maximizing call completion rate with $n = 4$, $c = 6$. 43
5.5. Experimental class-wise system parameters and comparative relation of the class-wise optimal performance measures under variable weighting factor with $n = 4$, $c = 6$. 48

5.6. Experimental class-wise system parameters and comparative relation of the class-wise optimal performance measures under different mobility of the calls with $n = 4, \omega = 10.0, c = 6$. 49
5.7. Experimental class-wise system parameters and comparative relation of the class-wise performance measures under different arriving rate with $n = 4, \omega = 10.0, c = 6$. 49
5.8. Experimental class-wise system parameters and comparative relation of the class-wise performance measures under different service rate with $n = 4, \omega = 10.0, c = 6$. 50
5.9. Performance evaluation of a two-class ring system by optimizing weighted call incompletion rate under variable weighting factor with $n = 4, \lambda_1 = \lambda_2 = 0.2, \mu_1 = \mu_2 = 1.0, \alpha_1 = 0.0, \alpha_2 = 10.0, c = 6$. 50
5.10. Performance evaluation of a two-class ring system by optimizing weighted call incompletion rate under variable weighting factor with $n = 4, \lambda_1 = \lambda_2 = 0.4, \mu_1 = \mu_2 = 1.0, \alpha_1 = 0.0, \alpha_2 = 10.0, c = 6$. 51
5.11. Performance evaluation of a two-class ring system by optimizing weighted call incompletion rate under different mobility of class 2 calls with $n = 4, \lambda_1 = \lambda_2 = 0.2, \mu_1 = \mu_2 = 1.0, \alpha_1 = 0.0, \omega = 10.0, c = 6$. 51
5.12. Performance evaluation of a two-class ring system by optimizing weighted call incompletion rate under different mobility of class 2 calls with $n = 4, \lambda_1 = \lambda_2 = 0.4, \mu_1 = \mu_2 = 1.0, \alpha_1 = 0.0, \omega = 10.0, c = 6$. 52
5.13. Performance evaluation of a two-class ring system by optimizing weighted call incompletion rate under different mobility of class 1 calls with $n = 4, \lambda_1 = \lambda_2 = 0.2, \mu_1 = \mu_2 = 1.0, \alpha_2 = 5.0, \omega = 10.0, c = 6$. 52
5.14. Performance evaluation of a two-class ring system by optimizing weighted call incompletion rate under variable arriving rate of class 1 calls with $n = 4, \lambda_2 = 0.2, \mu_1 = \mu_2 = 1.0, \alpha_1 = 1.0, \alpha_2 = 10.0, \omega = 10.0, c = 6$. . . 53
5.15. Performance evaluation of a two-class ring system by optimizing weighted call incompletion rate under variable arriving rate of class 2 calls with $n = 4, \lambda_1 = 0.2, \mu_1 = \mu_2 = 1.0, \alpha_1 = 0.0, \alpha_2 = 10.0, \omega = 10.0, c = 6$. . . 53

5.16. Performance evaluation of a two-class ring system by optimizing weighted call incompletion rate under variable arriving rate of class 2 calls with $n = 4$, $\lambda_1 = 0.2$, $\mu_1 = \mu_2 = 1.0$, $\alpha_1 = 1.0$, $\alpha_2 = 10.0$, $\omega = 10.0$, $c = 6$. . . 54

5.17. Performance evaluation of a two-class ring system by optimizing weighted call incompletion rate under variable service rate of class 1 calls with $n = 4$, $\lambda_1 = 0.1$, $\lambda_2 = 0.2$, $\mu_2 = 1.0$, $\alpha_1 = 0.0$, $\alpha_2 = 10.0$, $\omega = 10.0$, $c = 6$. 54

5.18. Performance evaluation of a two-class ring system by optimizing weighted call incompletion rate under variable service rate of class 1 calls with $n = 4$, $\lambda_1 = 0.2$, $\lambda_2 = 0.2$, $\mu_2 = 1.0$, $\alpha_1 = 5.0$, $\alpha_2 = 5.0$, $\omega = 10.0$, $c = 6$. 55

5.19. Performance evaluation of a two-class ring system by optimizing weighted call incompletion rate under variable service rate of class 1 calls with $n = 4$, $\lambda_1 = 0.2$, $\lambda_2 = 0.2$, $\mu_2 = 1.0$, $\alpha_1 = 1.0$, $\alpha_2 = 5.0$, $\omega = 10.0$, $c = 6$. 55

5.20. Performance evaluation of a two-class ring system by optimizing weighted call incompletion rate under variable service rate of class 1 calls with $n = 4$, $\lambda_1 = 0.1$, $\lambda_2 = 0.2$, $\mu_2 = 1.0$, $\alpha_1 = 5.0$, $\alpha_2 = 1.0$, $\omega = 10.0$, $c = 6$. 56

Chapter 1

Introduction

The rapid expansion of the mobile cellular market over the past few years has made apparent the inability of current wireless technology to meet future service requirements. The primary problem is the limited frequency bandwidth. If the frequency bandwidth remains unchanged, then the challenge for cellular engineers will be to meet customer demand with limited resources. Therefore, the use of the existing resources must be optimized.

Owing to the insufficiency of frequency bandwidth available for mobile communication systems, coverage areas are divided into contiguous regions which are called cells. When a call moves from one cell to another, it will request a channel from the new cell. This is called handoff. Due to the scarcity of channels the new cell may not provide the incoming call with a channel. As a result that handoff will be dropped. But from reliability of service point of view, this kind of situation is not desirable. So, special actions need to be taken so that customers may not lose their conversations while handoffs are in progress. This kind of preferential treatment can be provided by a call admission policy.

1.1 Call Admission Policies

Call admission policy is the set of decisions about when a new call will be served with a channel and when an existing call will be denied a handoff from one cell to another. The simplest form of all policies is "admit all policy (AAP)"– a call will be denied if and only if there is no channel. This is referred as a "complete sharing" policy for an ISDN environment in [Kau81].

On the other hand we can assume that access requests or handoffs can be regulated

by an intelligent controller which has the ability to deny the call even when there is one (or more than one) channel is available. The purpose of rejecting the call is to optimize average revenue, utilization of resources, etc which are defined as the system objectives. The set of policies which will maximize a particular system objective will be called "optimal policy (OP)". A policy that provides less than optimal performance will be called a "sub-optimal policy (SP)".

1.2 Single-Class and Two-Class Systems

A call class will be identified by its arrival rate, holding time and handoff rate. If all the calls in the system have same kind of stochastic properties then the system will be called *single-class cellular system*. A highway cellular network or a city cellular network can be considered as two different single-class systems. If each cell in the system supports the same two classes of calls then the system will be called *two-class cellular system*. In a two-class system, both classes will differ in their relative call arriving rate, relative service rate or relative handoff rate. We can consider a two-class system where pedestrians and customers in cars are two different classes or perhaps voice and data are two different classes.

1.3 Channel Allocation Schemes

- *Fixed Channel Allocation (FCA):* Each cell is assigned with a fixed number of channels.

  Most cellular networks today use this channel allocation scheme. System performance (call blocking probability) is measured using the Erlang-B model where voice calls arrive as a Poisson process and have exponential distributed service times. Each cell is considered an $M/M/c/c$ queueing system with c available channels.

- *Dynamic Channel Allocation (DCA);* All channels are available in every cell. Frequencies are assigned to new calls in such a way as to avoid interference with frequencies used in other cells (co-channel interference).

System performance is generally measured by simulation or Erlang-B approximation method. Simulation studies in [ESG82], [CR72], [And73], [KG78] have shown that existing DCA outperforms FCA at low to normal load. But at high load the system performance with DCA is no better than that with FCA and it can be worse, depending on the particular algorithm employed. This situation happens because the use of the channels is not optimized, DCA assigns the usable channel where it can, rather than where it should reconfigure the channels to achieve system optimal performance at high load.

DCA can be operated by a central switch which has the global information of the network or by a distributed control system, where decisions are made by the mobile and/or bases rather than centrally. Several channel allocation strategies for DCA with central control are available. Among them Maximum Packing, MP is the most popular and extensively studied [Ray], [EM83], [EM89]. The MP is analytically tractable and also in some sense optimal. Because the MP assigns the actual number of channels necessary to support the calls present at any time. This strategy rejects a call only when there is no possible way to allocate or reallocate a channel to that call. In other words, when a call finds the system state where the call would be normally blocked or dropped, but if the system finds that it is possible to rearrange the channels in use in any way to accept that call, subject to the re-use constraints, then the call will be accepted. That is the reason that the MP gives good performance at low to normal load and behaves like FCA at high load, see [EM83] and [EM89], where results are obtained by a Monte Carlo simulation. In [Ray], an algorithm with a limited number of channel rearrangements is shown to implement MP for a linear cellular system where one cell is used as a channel re-use constraints. Some partial algorithms are proposed in [SKK80]. In [Ray], Clique Packing, CP and Markov Allocation, MA are also studied as channel allocation strategies for DCA with central control. The CP is found as a impractical method, because it violates the channel re-use constraints and the MA is easy to implement and gives good performance.

In [CFI], a study is performed on DCA with distributed control. Here two algorithms are highlighted. The Timid Algorithm, where a mobile takes a channel only if there are no interferers. The Aggressive Algorithm, where the mobile may take a channel if there is interference is present and this algorithm is performed by call reconfigurations. The experimental results are obtained by Erlang-B approximation method, together with simulation. Those results suggest that the Aggressive Algorithm provides significant improvement in the system performance, while the Timid Algorithm is favorably comparable with FCA. Here it is found that MP is not a practically suitable strategy for microcellular systems. In this connection, an algorithm called Persistent Polite Algorithm, PPA which is a combination of the Timid Algorithm and the Aggressive Algorithm, is suggested.

1.4 Previous Work & Motivation

Call admission policies for multiple class queues have been widely studied before. In [GS83], the policy-iteration technique of MDP [How60] was used to find out optimal blocking policy for a multiple class arrivals. Ultimately it was concluded that the policy-iteration technique was computationally infeasible and sub-optimal "coordinate convex" policies that can be easily evaluated were proposed.

In [FG83], the optimal coordinate convex policies and its properties for multiple class of computer jobs sharing the same resources, such as memory or processors, were found. The computational aspects of finding optimal policies were not addressed. The geometrical structure of optimal policies for two or three different types of jobs was shown.

In [RT89a], the network-access problem of maximizing utilization subject to constraints on the probability of blocking was studied using the value-iteration algorithm of a semi-Markov decision process (SMDP) and Linear Programming (LP) algorithms. The main purpose of that paper was to illustrate the usefulness of SMDP in order to find the optimal design of telecommunication systems which have a dynamic nature (routing, flow control, integrated access etc). In [RT89b], it was observed that optimal coordinate convex policies provide good performance and are simple in structure.

In [RY93], the optimal call admission policy for a single cell of a cellular mobile network was studied. The policy-iteration method of MDP was used as a policy optimization technique. In that study, two distinct and independent classes of calls were considered in the system and the effectiveness of call admission in the context of i) optimizing weighted blocking criteria and ii) maximizing the revenue of the system were evaluated.

The optimistic empirical results of [RY93] has led us to extend this problem for a larger cellular system (both linear and ring) which supports single-class as well as two-classes of customers. In particular, the multiple cell system permits us to examine the impact of channel re-use constraints. In addition, our study has focussed mainly on optimizing weighted blocking criteria.

1.5 Model of the Cellular System

To perform a quantitative study of the optimal call admission policy for the cellular network, several assumptions are used. Arrivals of new calls of each class are independent Poisson processes. The duration of a call has an exponential distribution. The channel holding time of a call in a cell before a channel (handoff) from an adjacent cells is also exponentially distributed. Handoff rate is viewed as a measure of the mobility of a call. That is, a call of a highly mobile user will have a high handoff rate.

Premature termination of a call will happen either by blocking or handoff dropping. We do not consider the situation where a call is terminated due to radio propagation fading. Blocked new calls and dropped handoffs are cleared.

Co-channel interference is modeled by its simple re-use constraints in which the same channel can not be used in physically adjacent cells. As a channel allocation scheme, Maximum Packing DCA is used. In this scheme, in any cell an incoming call will be blocked or dropped by the admit all policy, if the sum of the active calls of that cell and any one of its adjacent cells equals the number of available channels.

1.6 Outline of this Work

This work explores where an optimal call admission policy helps to improve the system performance and where it does not. At the same time, we try to analyze why the system blocks a new call and drops a handoff when the system has sufficient resources. This study has also investigated the use of call admission policies to reduce unacceptable handoff dropping. For the two-class system, a comparative study is performed between two classes of distinct and independent calls. Finally, setting relative thresholds for new call blocking and handoff dropping have been delved into.

Markov decision process [How60], [Tij86] has been used throughout as a technique for searching for optimal policies.

Chapter 2

Markov Decision Processes (MDP)

2.1 Introduction to MDP

The Markov decision model is a versatile and powerful tool to analyze a probabilistic sequential process with an infinite planning horizon. This is an offshoot of the Markov model and dynamic programming. In fact, the latter is a recursive procedure for calculating optimal value functions from a functional equation. This functional equation is obtained from the *"Principle of Optimality"* [Bel57] which states that an optimal policy has the property that whatever the initial state and decisions are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first transition—the principle is always valid when the number of states and the number of actions are finite.

In 1960, Howard used the principle of optimality with Markov chain theory and developed a value-iteration algorithm and a policy-iteration algorithm, [How60]. He proposed the policy-iteration method for the solution of the long-duration problems. If N is the number of states of the system, then each iterative policy-iteration algorithm step requires solving N linear equations. To solve N linear equations requires a matrix of size $N \times N$ and $O(N^3)$ operations for matrix inversion. We can easily realize how fast the computational complexity and storage requirements grow with the increase of the number of states. For a four cells ring system where each cell supports two types of arrivals and the number of channels is 4, the number of states, $N = 2698$. In this case, on a typical workstation the policy-iteration algorithm becomes infeasible because of a computer-memory shortage. This difficulty was also apparent in [GS83] where policies for integrated services environment were studied.

In 1986, H.C Tijms developed a value-iteration algorithm for solving the continuous-time decision problem ,[Tij86]. It avoids the computationally burdensome procedure of solving systems of linear equations. It instead involves iterative computations and each iteration it generally performs $O(N^2)$ operations. However, if the number of transitions in each state is bounded then it performs only $O(N)$ operations. The storage requirement for the $N \times N$ matrix is not required. However, the number of iterations required by the value-iteration algorithm is typically problem dependent and usually increases with the number of states. The value-iteration algorithm was found to be feasible technique for solving Markov decision problems with a large number of states. The value-iteration algorithm provides upper and lower bounds on the long-run expected costs or gain rather than explicitly calculating the exact value of the gain like the policy-iteration algorithm. Tijms also modeled a modified value-iteration with a dynamic-relaxation factor to enhance the speed of convergence.

Before introducing the value-iteration algorithm of the Markov decision model for a discrete-time as well as a continuous-time dynamic system, we need to be familiar with the definitions of the system parameters. We shall also confine our discussion to discrete and continuous time Markov chains that are completely *ergodic.* At each step the system is classified into one of the possible number of states and subsequently a decision which is a set of actios, has to be chosen. This state is called the current state and denoted by x. If the set of all possible states is Λ, then $x \in \Lambda$. For state x, a set of decisions, $a(x)$ is given. For the sake of the *"Principle of Optimality"*, Λ and $a(x)$ are assumed to be finite. The set of all possible actions for the whole state space Λ is Π. As $a(x)$ is finite Π is also finite. A stationary policy $\hat{a} \in \Pi$ is a rule that always prescribes actions $a \in a(x)$ whenever the system is found in state x at a decision epoch. The consequences of the decision taken at the state x will be reflected in the future costs and this decision will not be affected by the past history of the system. We shall interpret $r(x, a)$ as the reward to be expected in the next transition out of state x if actions $a \in a(x)$ are chosen. At the next decision epoch, under actions a the system will be in state y with probability P_{xay}. The one step reward rate $r(x, a)$ and one step transition probability P_{xay} are assumed to be time homogeneous. For each stationary

policy $\hat{a} \in \Pi$, the stationary distribution will be $\{\pi_{\hat{a}}(x) : x \in \Lambda\}$. Then long-run average costs will be,

$$g(\hat{a}) = \sum_{x \in \Lambda} [\pi_{\hat{a}}(x) r(x, a)] \tag{2.1}$$

If we want to maximize the call completion rate of the system, then a stationary policy a^* will be said as optimal policy if $g(a^*) = g^*$, where $g^* = \max_{\hat{a} \in \Pi} g(\hat{a})$.

2.2 The Value-Iteration Algorithm for Discrete-Time Markov Decision Model

This method computes iteratively a sequence of value functions approximating the optimum long term average costs. The value functions provide lower and upper bound on the optimum average costs. Let $V_m(x)$ is interpreted as the total expected costs with m periods left to the time horizon when current state is x and $V_0(y)$ is interpreted as the terminal costs if the system ends up at state y. The value-iteration algorithm iteratively computes for $m = 1, 2, \ldots$, the value function $V_m(x)$ from

$$V_m(x) \quad = \quad \max_{a \in a(x)} \{r(x, a) + \sum_{y \in \Lambda} P_{xay} V_{m-1}(y)\} \qquad x \in \Lambda \tag{2.2}$$

starting with a value, $V_0(x)$, for each $x \in \Lambda$, which is naturally specified by the problem. The interpretation of $V_m(x)$ suggests that for large m the one step difference $V_m(x) - V_{m-1}(x)$ will be very close to optimum average costs and the stationary policy whose actions will maximize the right side of the equation 2.2 for all $x \in \Lambda$ will be very similar to the optimum policy. There is a useful theorem in this regard. For proof see p.191 [Tij86].

Theorem 1 *Suppose at time $m \geq 1$ the stationary policy $\hat{a}$ whose action maximizes the right side of the equation 2.2 for all $x \in \Lambda$. Then*

$$n_m \leq g(\hat{a}) \leq g^* \leq N_m \tag{2.3}$$

where

$$\begin{aligned} n_m &= \min_{x \in \Lambda} \{V_m(x) - V_{m-1}(x)\} \\ N_m &= \max_{x \in \Lambda} \{V_m(x) - V_{m-1}(x)\} \end{aligned}$$

step 0 : For all $x \in \Lambda$, choose $V_0(x) = q$; where $0 \leq q \leq \max_a r(x,a)$.
Set, $m = 1$.
step 1 : Compute the value of $V_m(x), x \in \Lambda$, from
$V_m(x) = \max_{a \in a(x)}[r(x,a) + \sum_{y \in \Lambda}\{P_{xay} V_{m-1}(y)\}]$
Determine $\hat{a}$ as the stationary policy whose actions maximize the above equation, for all $x \in \Lambda$.
step 2 : i) Compute,
$n_m = \min_{x \in \Lambda}\{V_m(x) - V_{m-1}(x)\}$
$N_m = \max_{x \in \Lambda}\{V_m(x) - V_{m-1}(x)\}$
ii) If $0 \leq N_m - n_m \leq \epsilon n_m$ then stop.
Otherwise, go to next step.
step 3 : $m = m + 1$, go to step 1.

Figure 2.1: Value-iteration algorithm for a discrete-time Markov chain.

.

Moreover, the sequence $\{n_m; m \geq 1\}$ *is non-decreasing and the* $\{N_m; m \geq 1\}$ *is non-increasing.*

2.2.1 Value-Iteration Algorithm

Equation 2.2 forms the basis of the value-iteration algorithm shown in Figure 2.1. Here, ϵ is prespecified tolerance number (say, $\epsilon = 0.001$).

In [Bat73] and in [SF79], it was proved that if Markov chain is *aperiodic* then $\lim_{m\to\infty}\{V_m(x) - mg^*\}$ is finite for all $x \in \Lambda$. That implies, $\lim_{m\to\infty} n_m = \lim_{m\to\infty} N_m = g^*$, this relation is independent of the initial state. In [Zij86], it was proved that the rate of the convergence of the bound of step 2 is geometric. This result is true for the aperiodic Markov chain if the optimal policy for average costs is searched. The number of iterations is not independent of the number of the states. This algorithm is not as robust as the policy-iteration algorithm. Because the policy-iteration algorithm convergences exponentially to the expected maximum costs and the number of iterations are independent of the number of states and the initial policy. The number of iterations of the value-iteration algorithm is problem dependent and increases with the increase of the number of states, [Tij86]. It will also depend on the tolerance number, ϵ and the initial value of $V_0(x)$, $x \in \Lambda$.

step 0 : For all $x \in \Lambda$, choose $V_0(x) = q$; where $0 \leq q \leq \max_a r(x,a)$.
Set, $m = 1$.

step 1 : For all $x \in \Lambda$, compute the value of $V_m(x)$,
$V_m(x) = \max_{a \in a(x)}\{r(x,a) + \sum_{y \in \Lambda}[P_{xay} V_{m-1}(y)]\}$
Determine $\hat{a}$ as the stationary policy whose actions maximize the above equation, for all $x \in \Lambda$.

step 2 : i) Compute,
$w = \arg\min_{x \in \Lambda}\{V_m(x) - V_{m-1}(x)\}$
$z = \arg\max_{x \in \Lambda}\{V_m(x) - V_{m-1}(x)\}$
$n_m = \min_{x \in \Lambda}\{V_m(x) - V_{m-1}(x)\}$
$N_m = \max_{x \in \Lambda}\{V_m(x) - V_{m-1}(x)\}$
ii) If $0 \leq N_m - n_m \leq \epsilon n_m$ then stop.
Otherwise, go to next step.

step 3 : Compute,
$\kappa = (N_m - n_m)$
$\nu = \sum_{y \in \Lambda}[\{P_{way} - P_{zay}\}\{V_m(y) - V_{m-1}(y)\}]$

step 4 : Compute the relaxation factor,
$\xi = \kappa / \{\kappa + \nu\}$

step 5 : For all $x \in \Lambda$, compute $V_m(x)$
$V_m(x) = V_{m-1}(x) + \xi\{V_m(x) - V_{m-1}(x)\}$

step 6 : $m = m + 1$, go to step 1.

Figure 2.2: Modified value-iteration algorithm for a discrete-time Markov chain.

2.2.2 Modified Value-Iteration Algorithm

To enhance the pace of convergence of the value iteration algorithm, Tijms proposed for using a relaxation factor in the previous algorithm which would be selected dynamically. This changes the value-Iteration algorithm of Figure 2.1 to the modified value-iteration algorithm, see Figure 2.2.

The modified value-iteration algorithm is formulated in order to achieve the implicit hope that the difference between the new upper and lower bounds N_{m+1} and n_{m+1} will decrease quickly. The relaxation factor has modified the properties of the value-iteration algorithm. The relaxation factor, $\xi \geq 0$ always. So the dynamic relaxation factor should significantly increase the speed of convergence of the value-iteration algorithm. Basically, this algorithm works quite well. But theoretically, it is not guaranteed that this algorithm will converge. The bounds in equation 2.3 hold here but n_m and N_m lose their monotonic property.

2.3 Uniformized Continuous-Time Markov Chain

In the section 2.2, we have seen that the value-iteration algorithm avoids the computationally burdensome task of solving systems of linear equations and requires only iterative computations. When the number of transitions in each state is bounded then in each iteration, it performs $O(N)$ operations, where N denotes the number of states. The value-iteration algorithm increases computational feasibility with the increase of the system size.

The value-iteration algorithm computes recursively for $m = 1, 2, \ldots$ the total expected costs, $V_m(x)$. But continuous-time Markov chain models allow for state transitions to occur at any real-valued time instant. Hence, we need to convert the continuous-time model into an equivalent discrete-time model. In this regard, we shall now discuss the powerful uniformization technique. The uniformization technique transforms a continuous-time Markov chain with nonuniform transition rates to a continuous-time Markov chain where transition rates are uniform.

Let, $\{X(t)\}$ be a continuous-time Markov chain which is characterized as follows:

The infinitesimal transition rates from state $x \in \Lambda$ to state $y \in \Lambda$ is λ_{xy}, where $x \neq y$. The amount of time until the next epoch (transition) occurs is exponentially distributed with mean $1/\theta(x)$, where $\theta(x) = \sum_{y \neq x} \lambda_{xy}$. If P_{xy} is the probability of the next state y given that current state is x, then,

$$P_{xy} = \begin{cases} \lambda_{xy}/\theta(x) & x, y \in \Lambda \text{ and } y \neq x \\ 0 & \text{otherwise} \end{cases}$$

Now, we consider the continuous-time Markov chain $\{\overline{X}(t)\}$ with the same state space, Λ. This chain is characterized as follows:

The infinitesimal transition rates from state x to state y is λ_{xy}, where $x, y \in \Lambda$. For each state x, the next epoch (transition) is generated by a Poisson process with rate θ, where $\theta \geq \theta(x)$, for all $x \in \Lambda$. In state x the process actually leaves at rate $\theta(x)$; but this equivalent to supposing that transitions occur at rate θ where only a fraction $\theta(x)/\theta$ are real transitions out of state x and the rest are fictitious transitions that leave the process in state x. If $\overline{P}_{xy}$, $x, y \in \Lambda$ is the one-step probability of the next state y

given that current state is x, then,

$$\overline{P}_{xy} = \begin{cases} \lambda_{xy}/\theta & y \neq x \\ \{1 - \theta(x)/\theta\} & y = x \\ 0 & \text{otherwise} \end{cases} \tag{2.4}$$

The *fictitious* transitions should have no effect on the behavior of the chain. By the memoryless property of the exponential distribution, it can be realized that the Markov process $\{\overline{X}(t)\}$ with a uniform transition rate is stochastically the same as the original Markov process $\{X(t)\}$ with state-dependent transition rates. The transitions from state to state for the Markov process $\{\overline{X}(t)\}$ can be modeled as a discrete-time Markov chain with self-loop transitions. For details, see [Ros83], [Tij86].

2.4 The Value-Iteration Algorithm for Continuous-Time Markov Decision Model

The value-iteration algorithm and the modified value-iteration algorithm for the discrete-time can be easily extended to continuous-time model. By the uniformization technique the state to state transition of a continuous-time Markov chain can be modeled as the state to state transition of a discrete-time Markov chain. In the value-iteration algorithm the one-step transition probability P_{xay} will be changed by the one-step uniformized transition probability $\overline{P}_{xay}$ (see equation 2.4). Tijms proposed modified value-iteration algorithm is showed in Figure 2.3.

The characteristics of the modified value-iteration algorithm for continuous-time Markov chain will remain almost the same as that for discrete-time Markov chain. Now uniformized rate, θ is added as another factor influencing the convergence rate of the algorithm. If θ is large then the self transition rate will be high and the bound of step 2 will converge slowly.

step 0 : For all $x \in \Lambda$, choose $V_0(x) = q$; where $0 \leq q \leq \max_a r(x,a)$.
Set, $m = 1$.

step 1 : For all $x \in \Lambda$, compute the value of $V_m(x)$
$V_m(x) = \max_{a \in a(x)}\{r(x,a) + \sum_{y \in \Lambda}[\overline{P}_{xay} V_{m-1}(y)]\}$
Determine $\hat{a}$ as the stationary policy whose actions maximize the above equation, for all $x \in \Lambda$.

step 2 : i) Compute,
$w = \arg\min_{x \in \Lambda}\{V_m(x) - V_{m-1}(x)\}$
$z = \arg\max_{x \in \Lambda}\{V_m(x) - V_{m-1}(x)\}$
$n_m = \min_{x \in \Lambda}\{V_m(x) - V_{m-1}(x)\}$
$N_m = \max_{x \in \Lambda}\{V_m(x) - V_{m-1}(x)\}$
ii) If $0 \leq N_m - n_m \leq \epsilon n_m$ then stop.
Otherwise, go to next step.

step 3 : Compute,
$\kappa = (N_m - n_m)$
$\nu = \sum_{y \in \Lambda}[\{\overline{P}_{way} - \overline{P}_{zay}\}\{V_m(y) - V_{m-1}(y)\}]$

step 4 : Compute the relaxation factor,
$\xi = \kappa / \{\kappa + \nu\}$

step 5 : For all $x \in \Lambda$, compute $V_m(x)$
$V_m(x) = V_{m-1}(x) + \xi\{V_m(x) - V_{m-1}(x)\}$

step 6 : $m = m + 1$, go to step 1.

Figure 2.3: Modified value-iteration algorithm for a continuous-time Markov chain.

Chapter 3

Markov Decision Process for a Cellular Network

3.1 Introduction

Markov decision process, MDP has computational complexity that increases with the increase of the size of the stochastic system. The size of the cellular network will depend on the capacity of the system, the number of cells and the number of classes of customers. In the first part of the chapter, it will be observed that efficient use of MDP is computationally feasible to find out the best policy for fairly large cellular network. In the second part, we shall see how that algorithm helps us to provide some pre-experimental information about the optimal policy.

3.1.1 Chapter Outline

In [RT89a], a technique to decompose the iterative equation of value-iteration algorithm when the reward rate does not depend on the action was showed. This kind of decomposition of the iterative equation remarkably enhances the speed of the value-iteration algorithm. To express the decomposition technique (with an extended version), we consider an arbitrary linear cellular system.

The System Parameters for an Arbitrary Linear Cellular System

In this chapter, we represent the current state of the system will be represented by a vector $x = (x_1, \ldots, x_n)$, where x_i is the number of calls in cell i and n is the number of cells in the system. The number of available channels is denoted by c. We have considered one cell as a channel re-use constraints.

We define clique sum $(i-1, i)$, $\sum_{(i-1,i)}$, as the sum of active calls in cell $i-1$ and

cell i. That is $\sum_{(i-1,i)} = x_{i-1} + x_i$. The set of feasible states, $\Lambda = \{x : \sum_{(k,k+1)} \leq c, k = 1, \ldots, n-1\}$.

When the system is at state x, a reject/accept decision must be made for each arrival and handoff in each cell. Each decision variable can be either 1 or 0, 1 stands for accept and 0 for reject. In state x, the decision variables for the cell i will be $a_i(x)$, $h_{i-1,i}(x)$ and $h_{i+1,i}(x)$, where

$$a_i(x) = \begin{cases} 1 & \text{if a new call is accepted at cell } i. \\ 0 & \text{if a new call is blocked at cell } i. \end{cases}$$

$$h_{i-1,i}(x) = \begin{cases} 1 & \text{if a handoff from cell } i-1 \text{ to cell } i \text{ is accepted.} \\ 0 & \text{if a handoff from cell } i-1 \text{ to cell } i \text{ is blocked.} \end{cases}$$

$$h_{i+1,i}(x) = \begin{cases} 1 & \text{if a handoff from cell } i+1 \text{ to cell } i \text{ is accepted.} \\ 0 & \text{if a handoff from cell } i+1 \text{ to cell } i \text{ is blocked.} \end{cases}$$

If there is any violation of any of the re-use constraints, the system will not provide a new channel to the new call or handoff, which means that call will be blocked or dropped immediately. So,

$$\begin{aligned} a_i(x) &= 0 \qquad \sum_{(i-1,i)} = c \text{ or } \sum_{(i,i+1)} = c \\ h_{i-1,i}(x) &= 0 \qquad \sum_{(i,i+1)} = c \\ h_{i+1,i}(x) &= 0 \qquad \sum_{(i-1,i)} = c \end{aligned}$$

So, under the admit all policy the decision variables for the cell i will be

$$a_i(x) = \begin{cases} 1 & \sum_{(i-1,i)} < c \text{ and } \sum_{(i,i+1)} < c \\ 0 & \text{otherwise} \end{cases}$$

$$h_{i-1,i}(x) = \begin{cases} 1 & x_{i-1} > 0 \text{ and } \sum_{(i,i+1)} < c \\ 0 & \text{otherwise} \end{cases}$$

$$h_{i+1,i}(x) = \begin{cases} 1 & x_{i-1} > 0 \text{ and } \sum_{(i-1,i)} < c \\ 0 & \text{otherwise} \end{cases}$$

In addition, to find a valid channel assignment scheme with above clique sums requirements, n can be even or odd for a linear system, but for a ring system, n should be even [Yen94].

Let us define the complement of the policies for cell i as $\overline{a_i(x)}$, $\overline{h_{i-1,i}(x)}$ and $\overline{h_{i+1,i}(x)}$. Where

$$\begin{aligned}
\overline{a_i(x)} &= 1 - a_i(x) \\
\overline{h_{i-1,i}(x)} &= 1 - h_{i-1,i}(x) \\
\overline{h_{i+1,i}(x)} &= 1 - h_{i+1,i}(x)
\end{aligned}$$

If at state x the action space (the set of all possible actions for the new call and handoff) for cell i is $a^i(x) = \{h_{i-1,i}(x), a_i(x), h_{i+1,i}(x)\}$, then the action space for the whole system at state x becomes, $a(x) = \{a^1(x), \ldots, a^i(x), \ldots, a^n(x)\}$. When the system is in state $x \in \Lambda$ and actions $a \in a(x)$ are chosen, the expected time until a new state is entered is given by $\tau(x,a) = [\sum_{i=1}^{n} a_i(x)\lambda_i + \sum_{i=1}^{n} x_i(\alpha_i + \mu)]^{-1}$. Actions a form a vector of $3n$ components and each component has two possibilities. So, the set, $a(x)$ contains 2^{3n} elements.

Let, $I = (1, \ldots, n)$ and e_i is a vector of zeroes, except for a one for the ith component, if P_{xay} , $y \in \Lambda$ is the probability of the next state is y given that current state is x and actions $a \in a(x)$ are chosen, then

$$P_{xay} = \begin{cases}
a_i(x)\lambda_i\tau(x,a) & y = x + e_i \\
\{\mu + [\overline{h_{i,i+1}(x)}p_i + \overline{h_{i,i-1}(x)}(1-p_i)]\alpha_i)\}x_i\tau(x,a) & y = x - e_i \\
h_{i,i+1}(x)x_i\alpha_i p_i\tau(x,a) & y = x + e_{i+1} - e_i \\
h_{i,i-1}(x)x_i\alpha_i(1-p_i)\tau(x,a) & y = x + e_{i-1} - e_i \\
0 & \text{otherwise}
\end{cases}$$

where $i \in I$ and λ_i is the call arriving rate in cell i in (calls/sec). We assume the call arrival process forms a stationary Poisson process. The average call holding time of a call is $1/\mu$ (sec/call). This call holding time is drawn from an exponential distribution. The average channel holding time of a call in cell i before requesting a channel (handoff) from its neighboring cells is $1/\alpha_i$ (sec/call). This channel holding time is also assumed as exponentially distributed continuous random variable. The probability that the call in cell i requesting a handoff goes to cell $(i+1)$ is p_i and $(1-p_i)$ equals the probability that the call in cell i requesting a handoff goes to cell $(i-1)$.

The indicator function is defined as

$$1[x > y] = \begin{cases} 1 & \text{when, } x > y \\ 0 & \text{otherwise} \end{cases}$$

MDP (Value-iteration algorithm)

The value-iteration algorithm is modeled for a continuous Markov-chain as follows (from previous chapter):

$$V_m(x) = \max_{a \in a(x)} \left[r(x,a) + \sum_{y \in \Lambda} \overline{P}_{xay} V_{m-1}(y) \right], \qquad x \in \Lambda. \tag{3.1}$$

where $\Gamma = [\sum_{i=1}^{n} \lambda_i + \max_{x \in \Lambda} \sum_{i=1}^{n} x_i(\alpha_i + \mu)]$ and Γ is uniformized rate for this problem.

The maximum total expected costs with m periods left to the time horizon when the current state is x is interpreted by $V_m(x)$. The uniformized transition probability is $\overline{P}_{xay}$ which is derived from P_{xay} and $\tau(x,a)$, see equation 2.4. To solve the recursive equation 3.1 at each iterative step requires us to search 2^{3n} possible candidate decisions. Ross and Tsang have modeled the value-iteration algorithm for finding the optimal policies in an ISDN environment which supports K-classes of calls. They have shown methods to decompose the value-iteration algorithm, that methods require K searches over two actions at each iterative step, when the reward rate does not depend on actions a.

In this problem, we use a generalization of the decomposition of Ross and Tsang. In particular, we can use the value-iteration algorithm to find the optimal policy by $3n$ searches over two actions in each iterative step, if the reward rate of each state, x, is a linear function of the action $a(x)$. We can write the reward rate for the current state x when actions a are chosen as

$$\begin{aligned} r(x,a) &= C(x) + \sum_{i=1}^{n} [c_{i,i-1}(x) h_{i,i-1}(x) + \\ &\quad c_{i,i}(x) a_i(x) + c_{i,i+1}(x) h_{i,i+1}(x)]. \end{aligned} \tag{3.2}$$

where $C(x)$ and $c_{i,j}(x)$, $j = i-1$, i, $i+1$ are the constants for the state x, they will be selected depending on the objective which is going to be optimized by the value-iteration algorithm. As for example, if we want to minimize the call incompletion rate

or maximize the negative call incompletion rate of this system, then the reward rate will be

$$\begin{aligned} r(x,a) &= \sum_{i=1}^{n}[-x_i\alpha_i(1-p_i)\overline{h_{i,i-1}(x)} - \lambda_i\overline{a_i(x)} - x_i\alpha_i p_i\overline{h_{i,i+1}(x)}] \\ &= -\sum_{i=1}^{n}(\lambda_i + x_i\alpha_i) + \sum_{i=1}^{n}[x_i\alpha_i(1-p_i)h_{i,i-1}(x) \\ &\quad +\lambda_i a_i(x) + x_i\alpha_i p_i h_{i,i+1}(x)] \end{aligned} \tag{3.3}$$

Comparing the equation 3.3 with equation 3.2 we can write,

$$C(x) = -\textstyle\sum_{i=1}^{n}(\lambda_i + x_i\alpha_i).$$

$$c_{i,i-1}(x) = x_i\alpha_i(1-p_i).$$

$$c_{i,i}(x) = \lambda_i.$$

$$c_{i,i+1}(x) = x_i\alpha_i p_i.$$

The equation 3.1 can be written as follows:

$$\begin{aligned} V_m(x) &= C(x) + V_{m-1}(x) + \max_{a\in a(x)}[\sum_{i=1}^{n}\{c_{i,i-1}(x)h_{i,i-1}(x) + c_{i,i}(x)a_i(x) \\ &\quad +c_{i,i+1}(x)h_{i,i+1}(x)\} + \sum_{y\in\Lambda}\overline{P}_{xay}\{V_{m-1}(y) - V_{m-1}(x)\}] \\ &= C(x) + V_{m-1}(x) + (1/\Gamma)\sum_{i=1}^{n}[x_i\mu\{V_{m-1}(x-e_i) - V_{m-1}(x)\} \\ &\quad +a_i(x)(\lambda_i\{V_{m-1}(x+e_i) - V_{m-1}(x)\} + \Gamma c_{i,i}(x)) \\ &\quad +h_{i,i-1}(x)(x_i\alpha_i(1-p_i)\{V_{m-1}(x+e_{i-1}-e_i) - V_{m-1}(x)\} \\ &\quad +\Gamma c_{i,i-1}(x)) + \overline{h_{i,i-1}(x)}x_i\alpha_i(1-p_i)\{V_{m-1}(x-e_i) - V_{m-1}(x)\} + \\ &\quad h_{i,i+1}(x)(x_i\alpha_i p_i\{V_{m-1}(x+e_{i+1}-e_i) - V_{m-1}(x)\} + \Gamma c_{i,i+1}(x)) + \\ &\quad \overline{h_{i,i+1}(x)}x_i\alpha_i p_i\{V_{m-1}(x-e_i) - V_{m-1}(x)\}] \end{aligned} \tag{3.4}$$

The decision variables a_i, $h_{i,i-1}$ and $h_{i,i+1}$ can be expressed in terms of the value function $V_m(x)$ as

$$a_i(x) = 1[V_{m-1}(x+e_i) + \{\Gamma/\lambda_i\}c_{i,i}(x) > V_{m-1}(x)] \tag{3.5}$$

$$h_{i,i-1}(x) = 1[V_{m-1}(x+e_{i-1}-e_i) + \{\Gamma/(x_i\alpha_i(1-p_i))\}c_{i,i-1}(x) > V_{m-1}(x-e_i)] \tag{3.6}$$

$$h_{i,i+1}(x) = 1[V_{m-1}(x+e_{i+1}-e_i) + \{\Gamma/(x_i\alpha_i p_i)\}c_{i,i+1}(x) > V_{m-1}(x-e_i)] \tag{3.7}$$

step 0 : Choose the reward rate, $r(x,a)$. If $r(x,a)$ is a linear function of a then go to next step, otherwise stop.

step 1 : For all $x \in \Lambda$, choose $V_0(x) = q$; where $0 \leq q \leq \max_a r(x,a)$. Set, $m = 1$.

step 2 : Decompose $V_m(x)$, where $V_m(x) = \max_{a \in a(x)}\{r(x,a) + \sum_{y \in \Lambda}[\overline{P}_{xay} V_{m-1}(y)]\}$

step 3 : For all $x \in \Lambda$, compute the value of $V_m(x)$, from the step 3. Determine $\hat{a}$ as the stationary policy whose actions maximize the equation of step 3, for all $x \in \Lambda$.

step 4 : i) Compute,
$w = \arg\min_{x \in \Lambda}\{V_m(x) - V_{m-1}(x)\}$
$z = \arg\max_{x \in \Lambda}\{V_m(x) - V_{m-1}(x)\}$
$n_m = \min_{x \in \Lambda}\{V_m(x) - V_{m-1}(x)\}$
$N_m = \max_{x \in \Lambda}\{V_m(x) - V_{m-1}(x)\}$
ii) If $0 \leq N_m - n_m \leq \epsilon n_m$ then stop.
Otherwise, go to next step.

step 5 : Compute,
$\kappa = (N_m - n_m)$
$\nu = \sum_{y \in \Lambda}[\{\overline{P}_{way} - \overline{P}_{zay}\}\{V_m(y) - V_{m-1}(y)\}]$

step 6 : Compute the relaxation factor,
$\xi = \kappa / \{\kappa + \nu\}$

step 7 : For all $x \in \Lambda$, compute $V_m(x)$
$V_m(x) = V_{m-1}(x) + \xi\{V_m(x) - V_{m-1}(x)\}$

step 8 : $m = m + 1$, go to step 4.

Figure 3.1: Decomposed modified value-iteration algorithm for a continuous-time Markov chain.

3.1.2 Decomposed Modified Value-iteration Algorithm

We can easily realize that the value-iteration algorithm with the new decomposed approach becomes very efficient. At each iteration, the number of evaluations searching for the optimal policy is reduced by the factor of $2^{3n}/n$. The value-iteration algorithm will be further accelerated by introducing the dynamic relaxation factor. With the combinations of these two, we were able to experimentally solve fairly large system. So our algorithm will be the combination of the decomposed recursive equation and the dynamic relaxation factor. We shall call this new algorithm the *decomposed modified value-iteration algorithm*, see Figure 3.1.

3.2 Structural Properties of the Optimal Policy

In this section, we explore the idea to observe the structural properties of the optimal policy under the minimization of weighted call incompletion rate. Here the weight, ω implies the relative penalty for dropping a handoff over blocking a new call. Minimization of the call incompletion rate or maximization call completion rate is equivalent to minimization of weighted call incompletion rate with weight, $\omega = 1.0$.

Theorem 2 *In order to minimize the weighted call incompletion reward rate (when $\omega > 1$), if at state x a new call at cell i is accepted, then at state $x + e_{i-1}$ a handoff from cell $i-1$ to cell i will be accepted. But the converse is not true.*

Proof: In this case the reward rate will be

$$\begin{aligned} r(x,a) &= \sum_{i=1}^{n}[-x_i\omega\alpha_i(1-p_i)\overline{h_{i,i-1}(x)} - \lambda_i\overline{a_i(x)} - x_i\omega\alpha_i p_i\overline{h_{i,i+1}(x)}] \\ &= -\sum_{i=1}^{n}(\lambda_i + x_i\omega\alpha_i) + \sum_{i=1}^{n}[x_i\omega\alpha_i(1-p_i)h_{i,i-1}(x) \\ &\quad +\lambda_i a_i(x) + x_i\omega\alpha_i p_i h_{i,i+1}(x)] \end{aligned} \tag{3.8}$$

Comparing the equation 3.8 with equation 3.2 we can write,

$$C(x) = -\textstyle\sum_{i=1}^{n}(\lambda_i + x_i\omega\alpha_i).$$
$$c_{i,i-1}(x) = x_i\omega\alpha_i(1-p_i).$$
$$c_{i,i}(x) = \lambda_i.$$
$$c_{i,i+1}(x) = x_i\omega\alpha_i p_i.$$

Using the above $c_{i,i}(x)$ in the optimal policy $a_i(x)$ of equation 3.5 we shall obtain,

$$a_i(x) = 1[V_{m-1}(x+e_i) + \Gamma > V_m(x)] \tag{3.9}$$

Using the above $c_{i,i-1}(x)$ in the optimal policy $h_{i,i-1}(x)$ of equation 3.6 we shall obtain,

$$h_{i,i-1}(x) = 1[V_{m-1}(x+e_{i-1}-e_i) + \omega\Gamma > V_m(x-e_i)] \tag{3.10}$$

For the state s, where $s = x + e_{i-1}$, the optimal policy $h_{i-1,i}(s)$ will be (from equation 3.10)

$$h_{i-1,i}(s) = 1[V_{m-1}(s+e_i-e_{i-1}) + \omega\Gamma > V_{m-1}(s-e_{i-1})].$$

If we rewrite the above policy $h_{i-1,i}(s)$ changing s by $x + e_{i-1}$, then

$$h_{i-1,i}(s) \quad = \quad 1[V_{m-1}(x + e_i) + \omega\Gamma > V_{m-1}(x)] \tag{3.11}$$

After accepting new a call in cell i at state x, the state $x + e_i$ will be entered. If this event is permitted by the optimal policy then the handoff from cell $i - 1$ to cell i at state s will be permitted definitely (comparing equation 3.9 and 3.11), that means the state $x + e_i$ will be entered too. But converse is not true. In other words, a new call may not be accepted in cell i at state x, but a handoff from cell $i - 1$ to cell i at state s may be accepted.

Let, $\phi \in \Lambda$ denotes a state which is entered after accepting a new call and $\upsilon \in \Lambda$ is a state which is entered after accepting a handoff. If Φ is the set of all states those are entered by accepting new calls and Υ is the set of all possible states those are entered by accepting handoffs, then in order to minimize weighted call incompletion rate (when $\omega > 1$), $\Phi \subset \Upsilon$. For minimizing weighted call incompletion rate (when $\omega < 1$), $\Upsilon \subset \Phi$. However, a reward function with $\omega < 1$ does not have any practical significance.

This theorem is true for any $\lambda_i > 0$, $\mu > 0$, $\alpha_i > 0$ and $p_i > 0$, no matter whether the cellular network is linear or ring. This result can be easily extended for any other structure of the network. It also holds for any set of interference constants.

Corollary 1 *In order to maximize the call completion rate or minimize the call incompletion rate ($\omega = 1.0$), a new call arriving at cell i at state x is admitted if a handoff from cell $(i - 1)$ to cell i at state $x + e_{i-1}$ is allowed.*

It is also true that $a_i(x) = h_{i+1,i}(x + e_{i+1})$. Here $\Upsilon = \Phi$. The above corollary reduces the total number of decision variables by the factor of 2/3. This reduction of decision variables is very significant when the system has a large number of cells and each cell supports K call classes of arrival, where K is large.

Chapter 4

Empirical Performance Measures for a Single-Class Cellular Network

4.1 Introduction

This chapter is a quantitative study on the effectiveness of the call admission policy in the context of a single-class linear and a ring cellular system. In linear cellular network, the cells are connected in a linear fashion (see fig. 4.1). If the two cells at the ends of the linear network are connected together then network will be called a ring cellular network (see fig. 4.2). A highway cellular network can be modeled as linear cellular network and each block of the city can be considered one small ring cellular network.

This study considers linear and ring cellular networks. We shall observe the effectiveness of the optimal policy under variable arrival rate, service rate and handoff rate of the calls while we are maximizing the call completion rate or minimizing weighted call incompletion rate.

Experiments in the chapter are performed in two stages: first, maximization of call completion rate, and second, minimization of weighted call incompletion rate.

4.1.1 Definitions of the System Parameters

Let λ denotes the call arrival rate (calls/sec) in each cell. In addition, let μ and α denote the service rate (calls/sec) and average handoff rate respectively. The number of cells in the system is denoted by n, and c represents the number of available channels. Offered load of the calls is $\rho = N(\lambda/\mu)$.

Figure 4.1: Highway Linear Cell Model, Ba= base station, n = number of the cells.

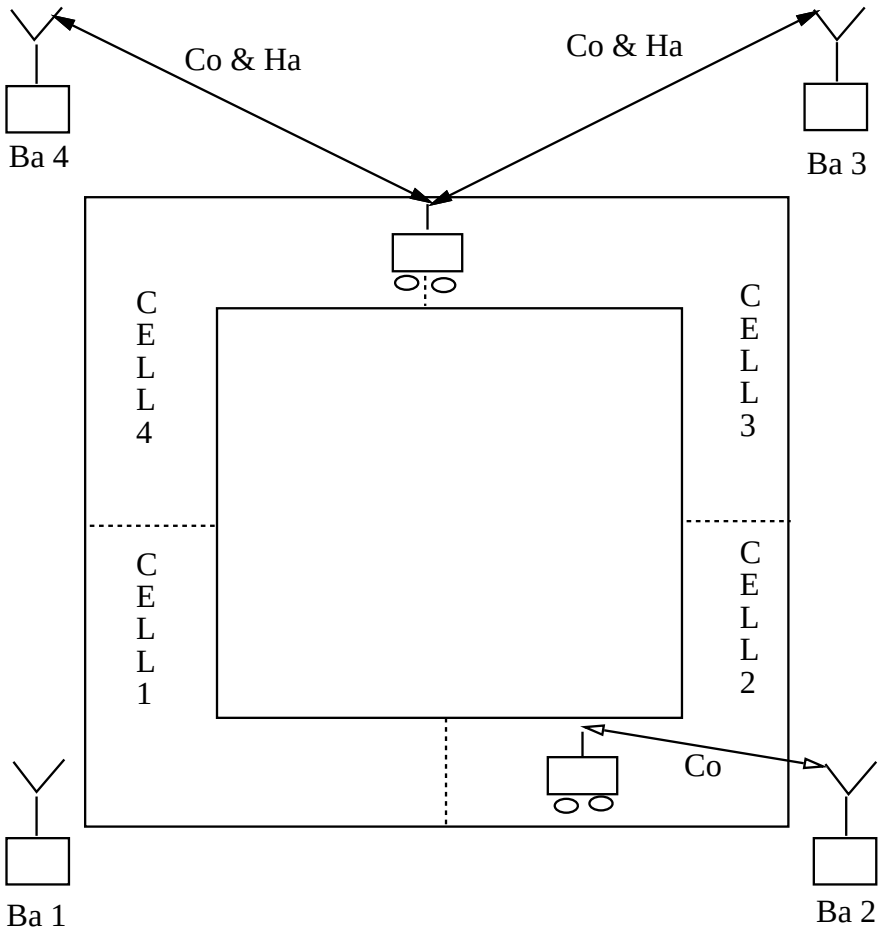

Figure 4.2: City Ring Cell Model, Ba= base station, Co= conversation, Ha= handoff.

4.1.2 Performance Measures

It is assumed that a premature termination of a call happens either by blocking a new call or dropping a call in progress. Probability of new call blocking is defined as the probability of a call is blocked at the entrance of system and probability of handoff dropping is the probability that a call will be dropped during the handoffs. For evaluating the system performance, probability of new call blocking and probability of handoff dropping will be measured.

Definition of the Parameters

Let cc^{π} denotes the average rate (calls/sec) of call completion under the policy π, where b^{π} and d^{π} are the average rate of new call blocking (calls/sec) and average rate of handoff droppings (calls/sec) respectively, see Figure 4.3.

Formulation of the Performance Measures

Let $N(t)$ denotes the number of new calls which will arrive at the system by time t, $b^{\pi}(t)$ are the number of calls that will be blocked by time t if the system uses the stationary policy π and $d^{\pi}(t)$ are the number of handoffs that will be dropped by time under the stationary policy π. Then,

The probability of new call blocking under the policy π will be

$$\begin{aligned} B^{\pi} &= \lim_{t\to\infty} \frac{b^{\pi}(t)}{N(t)} \\ &= \lim_{t\to\infty} \frac{b^{\pi}(t)/t}{N(t)/t} \\ &= \frac{b^{\pi}}{n\lambda} \end{aligned} \tag{4.1}$$

The probability of handoff dropping under the policy π will be

$$\begin{aligned} D^{\pi} &= \lim_{t\to\infty} \frac{d^{\pi}(t)}{N(t)} \\ &= \lim_{t\to\infty} \frac{d^{\pi}(t)/t}{N(t)/t} \\ &= \frac{d^{\pi}}{n\lambda} \end{aligned} \tag{4.2}$$

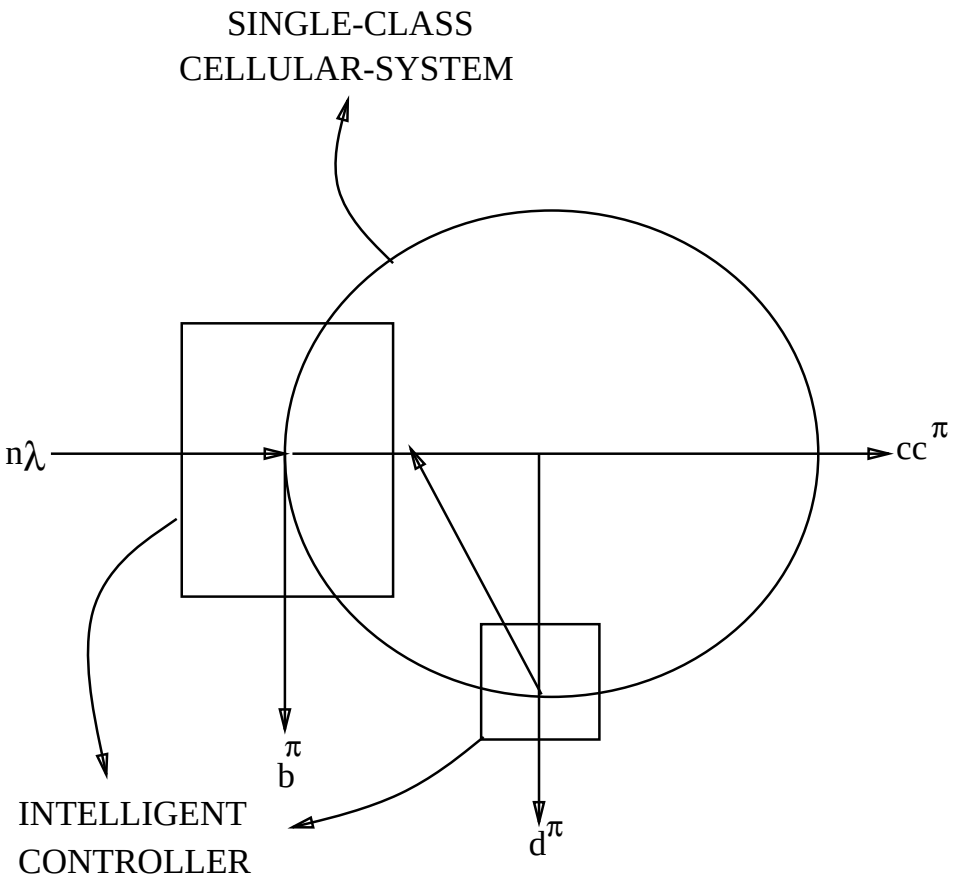

Figure 4.3: Average rate flow in a single-class mobile cellular system under the stationary policy is π.

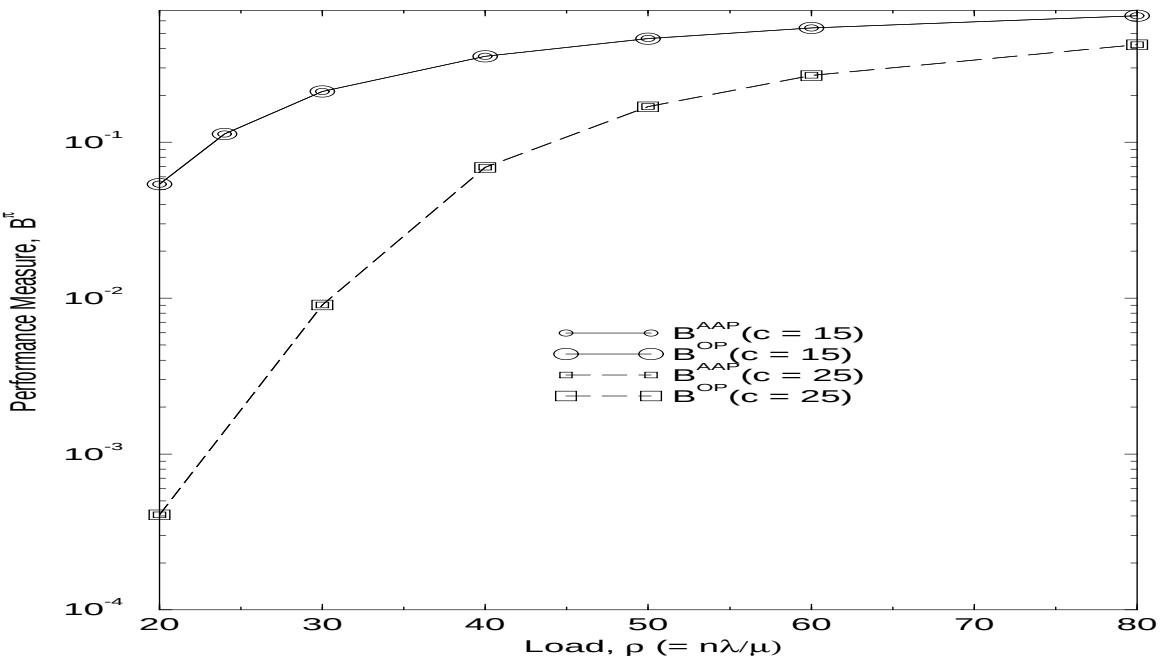

Figure 4.4: Performance evaluation of a ring system by maximizing call completion rate under variable load with $c = 15$ and $c = 25$, when $n = 4$.

4.2 Maximizing Call Completion rate

A single-class system can model the situation in users are stationary and calls always stay inside the same cell, for the whole duration of the call or users are mobile and generate handoffs. In this study, for a system with mobile calls, it is assumed that an arbitrary handoff of a cell can go either of the adjacents cells with equal probability.

Experiments in this section are performed in two stages. First, the effectiveness of maximizing call completion rate is observed in the context of stationary calls. Afterwards, we assume that calls are mobile and observe the effect of the maximizing call completion rate on the different performance measures.

4.2.1 Stationary Calls in the 4 Cell Ring System

For a ring system with $n = 4$ cells, probability of call blocking is plotted in Figure 4.4 as function of load. Two different values of c are used, $c = 15$, $c = 25$. However, no improvement of the system performance over the admit all policy is observed. In the ring system the position of each cell is symmetric and the arriving calls at each cell have the same stochastic properties. So, the system can not preferentially serve the calls of a

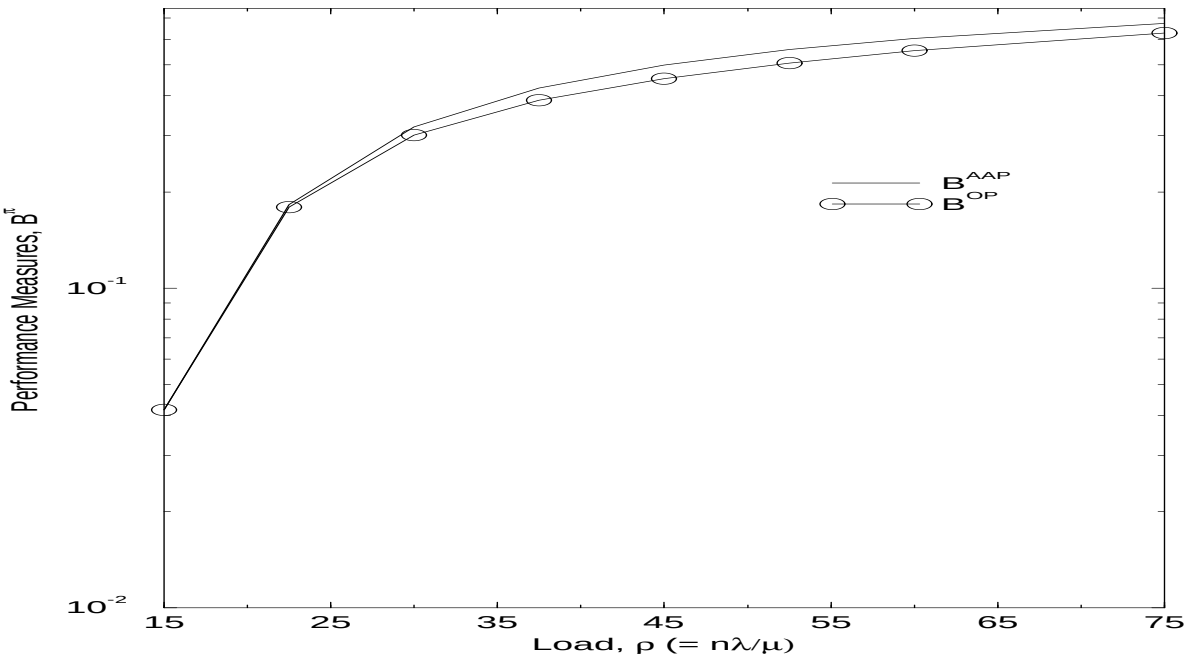

Figure 4.5: Performance evaluation of a linear system by maximizing the call completion rate under variable load with $n = 3$, $c = 15$.

particular cell by blocking the calls of the neighbors. Hence, the optimal policy can not significantly outperform the admit all policy. Experimentally, it is observed that at low load the admit all policy is optimal but at higher load the system does not like to spend the last channel of the system in a cell. Because that assignment of the last channel will make its neighbors run out of channels that will probably block more customers in near future. However, this saving of a channel does not significantly alter the overall system performance.

4.2.2 Stationary Calls in the 3 Cell Linear System

In Figure 4.5, probability of call blocking is shown at different loads for a linear system with $n = 3$. From Figure 4.5, we see that the optimal policy has a noticeable but not dramatic effect on the system performance. In the linear system with three cells, the optimal policy penalizes arrivals in the middle cell. If the system wants to support a new call in the middle cell then it needs to spend not only a channel from the middle cell but also a channel from the two cells at the ends. At higher load the system behaves conservatively towards this kind of expenditure. As a result the system blocks the new calls frequently at the middle cell and the optimal policy exhibits slightly better

performance over the admit all policy. At low load (when new call arriving rate is small) the system does not want to keep channels idle waiting for the calls at the end cells (because the aim of the system is maximizing call completion rate) and it provides channels to the calls in the middle cell. As a result the admit all policy becomes the optimal policy as load decreases.

If the system wants to support an incoming call at the end cells, then it needs to spend one channel which is shared by only one neighbor. So, supporting a call in the end cells is more efficient use of a channel than supporting a call in any arbitrary cells in the system of size n. As a result under the optimal policy the cells at the two ends never block a call (this happens not only for $n = 3$ but also any value of n) neither at higher load nor at lower load. This situation also implies that the cells at the two ends have the lowest probability of new call blocking among all cells in the system.

For the n cells linear system of Figure 4.1, we shall see that by the optimal policy, cells 2 and $n - 1$ block the calls more frequently. This happens because the optimal policy prefers to serve the calls in the end cells. If we consider the cells closer to the center, then it will be observed that those cells block new calls with almost equal probability by the optimal policy which will not significantly differ from the admit all policy. Because of the physical position, the blocking characteristics of those cells (closer to the center) are insignificantly affected by the preferential treatment which the optimal policy provides to the end cells. So, as the number of cells in the linear system increases, the improvement of the overall performance measure will decrease. This also implies why we did not observe any improvement for maximizing call completion rate in ring system, see Figure 4.4. A ring system of size four (each cell has two neighboring cells) can be considered as an infinitely long linear system where effects of the two end cells on the overall system performance will be negligible.

All of our following experiments will concentrate on a ring system of size four with the hope that the effectiveness of call admission policy for a ring system of size four will express the effectiveness of call admission policy in the context of an infinitely large linear system as well as a ring system of any size. In addition, a ring system of size four may represent the ring system of any size, because addition of a cell in the ring

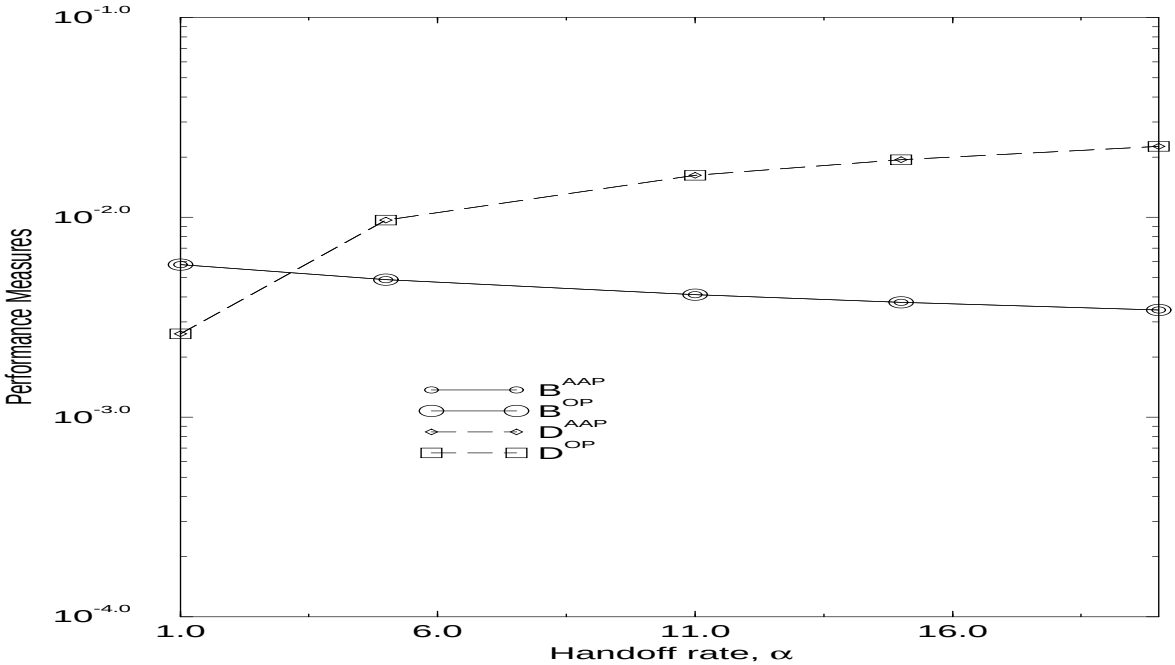

Figure 4.6: Performance evaluation of a ring system by maximizing call completion rate under variable handoff rate with $n = 4$, $\lambda = 2.5$, $\mu = 1.0$, $c = 12$.

system does not destroy the symmetry of the system. There are some other reasons for this consideration. In Chapter 3, we have described that for achieving a valid channel assignment scheme, when one cell is considered as a channel re-use constraint, then the number of cells in the ring system should be even. Four is the minimum number for the size of a ring system for which we can use any value of c and the clique sum requirements are sufficient to ensure a valid channel assignment scheme.

4.2.3 Mobile Calls

In Chapter 3, we have proved that when the system wants to maximize the call completion rate, then it has to treat handoffs and new calls equally. So, for maximizing call completion rate no preferential attention will be paid to either handoffs or new calls. Under this restriction, the system can not follow a policy which is very different from the admit all policy. As a result, in Figure 4.6 and Figure 4.7 where performance measures are plotted as a function handoff rate at two different loads of the system, we do not observe any significant improvement in the system performance when maximization of call completion rate is achieved. The same kind of empirical results are also observed in Figure 4.8 where performances are measured by varying call arriving

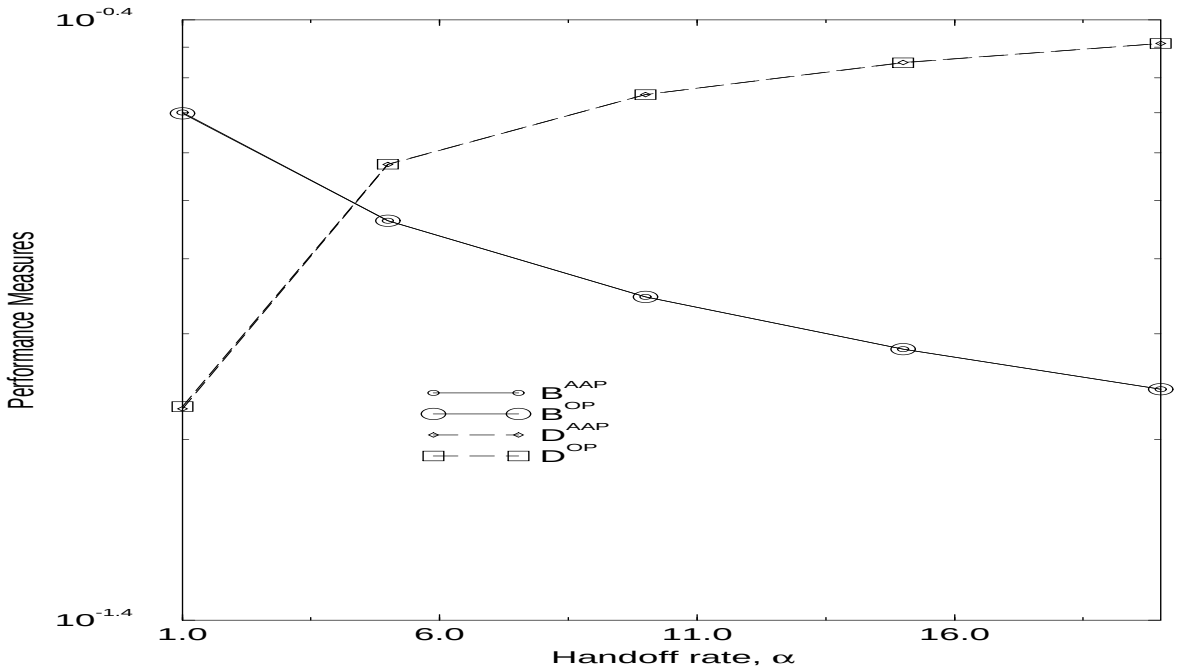

Figure 4.7: Performance evaluation of a ring system by maximizing call completion rate under variable handoff rate with $n = 4$, $\lambda = 7.5$, $\mu = 1.0$, $c = 12$.

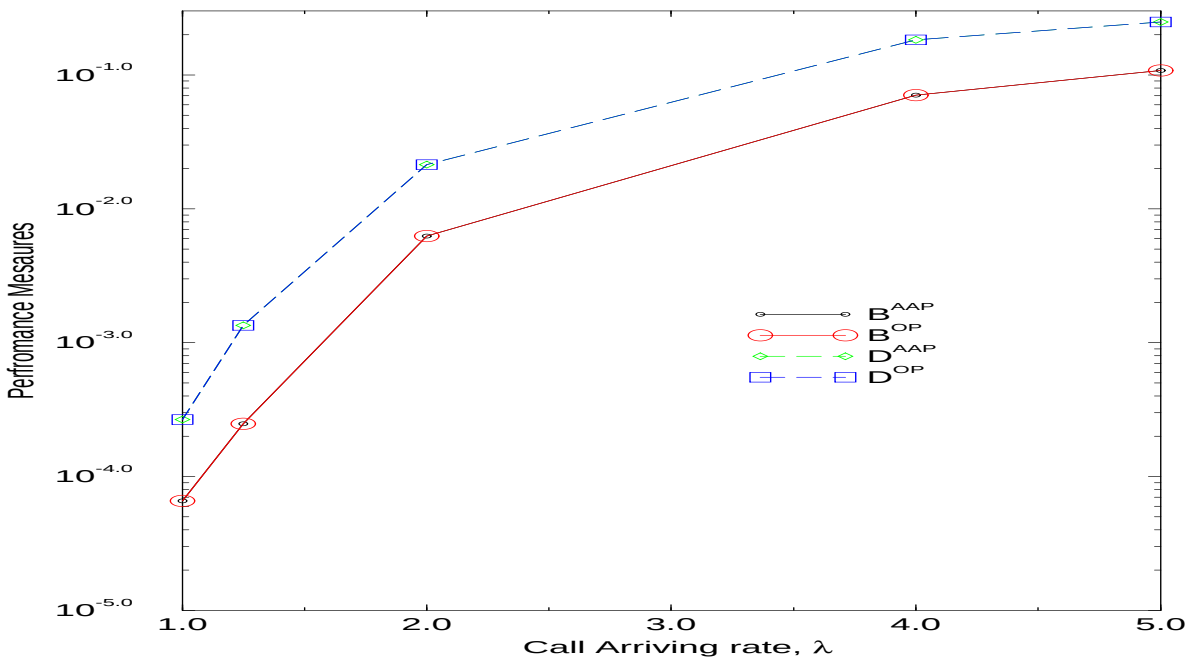

Figure 4.8: Performance evaluation of a ring system by maximizing call completion rate under variable call arriving rate with $n = 4$, $\mu = 1.0$, $\alpha = 10.0$, $c = 10$.

rate with $\mu = 1.0$, $\alpha = 10.0$ and $c = 10$.

4.2.4 Conclusion

We have observed, the policy which maximizes call completion rate never differs significantly from the admit all policy for any set of parameters of a single-class system. As a consequence we have seen that the probability of handoff dropping is high at higher load and at higher mobility or higher handoff rate, see Figure 4.6, Figure 4.7 and Figure 4.8.

Low probability of handoff dropping is defined as the reliability of the service. So, both at higher load and higher mobility the system loses its quality of service. However we can not minimize only the handoff dropping rate, because the probability of handoff dropping can be made zero by blocking every new calls. To handle this situation, we are interested to introduce a weighting factor, ω in the reward rate of minimizing call incompletion rate, where the value of ω will be always greater than 1; see Chapter 3, equation 3.8. The value of ω is physically interpreted as how many times more the system will be penalized for dropping a handoff than for blocking a new call.

4.3 Optimizing Weighted Reward rate

In this section, the weighted performance measures are plotted as a function of weight, handoff rate, call arriving rate and service rate.

4.3.1 Performance Measures with Variable Weighting Factor

As ω increases the relative penalty for dropping a handoff increases. So, it is natural that the system will decrease the probability of handoff dropping with the increase of ω. This will be achieved by saving more channels for the handoffs, and as a consequence the system will reject more new calls. The same results are observed in Figure 4.9 and Figure 4.10, where two experiments are performed by varying the weighting factor, ω at two different loads of the system. It is also observed that to save a fixed number of handoffs from dropping, the system will block higher number of calls.

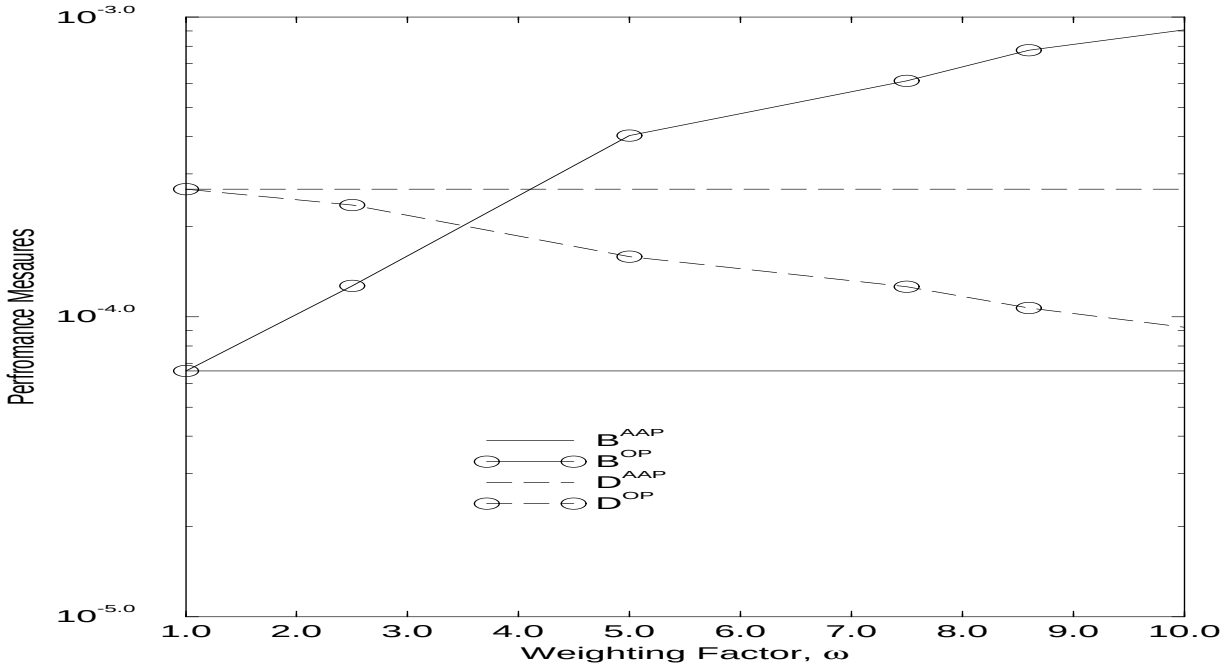

Figure 4.9: Performance evaluation of a ring system by optimizing weighted call incompletion rate under variable weighting factor with $n = 4$, $\lambda = 1.0$, $\mu = 1.0$, $\alpha = 10.0$, $c = 10$.

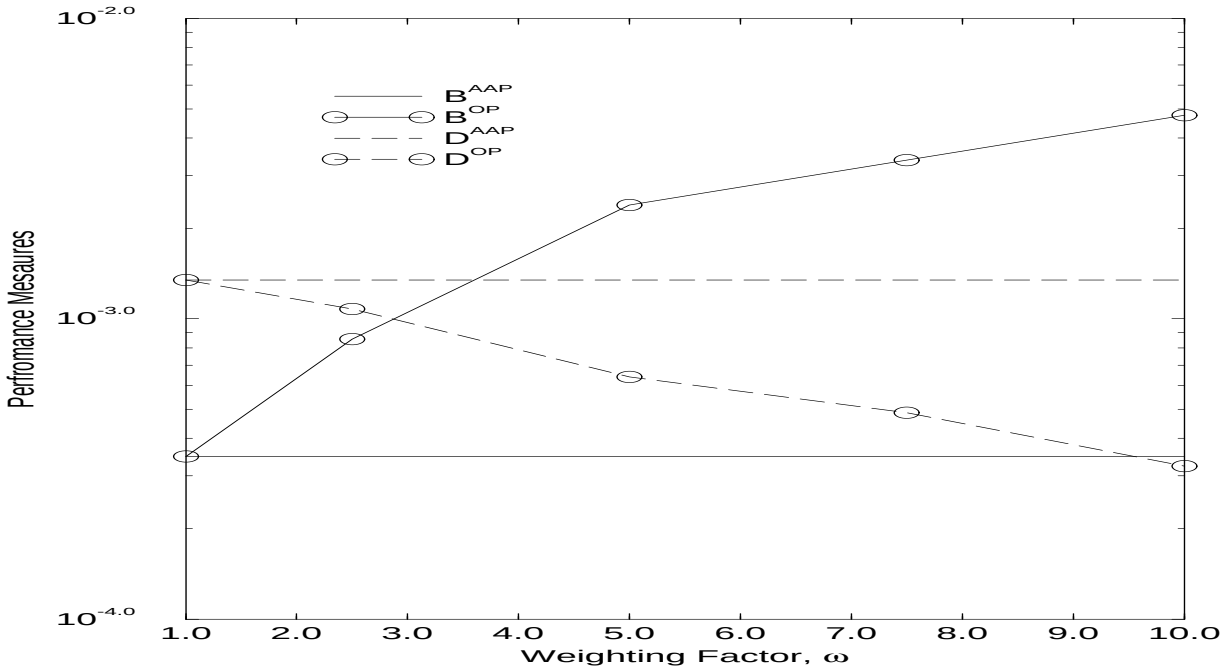

Figure 4.10: Performance evaluation of a ring system by optimizing weighted call incompletion rate under variable weighting factor with $n = 4$, $\lambda = 1.25$, $\mu = 1.0$, $\alpha = 10.0$, $c = 10$.

Purpose of the Weight

If the system parameters are constant, then by selecting a proper value of ω the system providers can be provided with any value of probability of handoff dropping. Selection of ω means selection of the stationary policy associated with it. In some places, like cities, where the load of the system fluctuates from time to time. It is possible to keep the same probability of handoff dropping through out the whole day by changing the value of ω with the change of the load. This kind of technique will keep the reliability of service at constant level and new calls will not suffer the penalty associated with the worst case load throughout the whole day. Otherwise, for maintaining good reliability of service the value of ω will be selected considering the time when the offered load is high which will always penalize new calls, unnecessarily when the offered load is low. This can be noted by comparing Figure 4.9 with $\lambda = 1.0$ and Figure 4.10 with $\lambda = 1.25$.

Selection of the Weight

Due to the discrete nature of the policy optimization, there may exist a range of weights those yield the same optimal policy and selection of the weight will not be unique. Then for the weight, any value from that range can be selected to obtain the system policy. Again, the optimal policy associated with a ω may not be unique. If $\omega - \delta$ and $\omega + \delta$, (where δ is very small positive number) provide two different stationary policies, then either of the two policies can be considered as optimal policy for ω.

4.3.2 Performance Measures with Different Handoff Rate

When the relative penalty associated handoff dropping is high (say, $\omega = 10$) and then the system will like to reserve channels for a call in nearby cells. A side effect of this kind of channel reservation for handoffs is that new calls will be rejected from the system with high probability. If speed increases, sufficient increase in the weight is needed to ensure that handoff dropping does not increase. So, as the value of ω remains fixed then it is natural that probability of handoff dropping will increase with the increase of handoffs rate. All of these consequences are observed in Figure 4.11 and Figure 4.12,

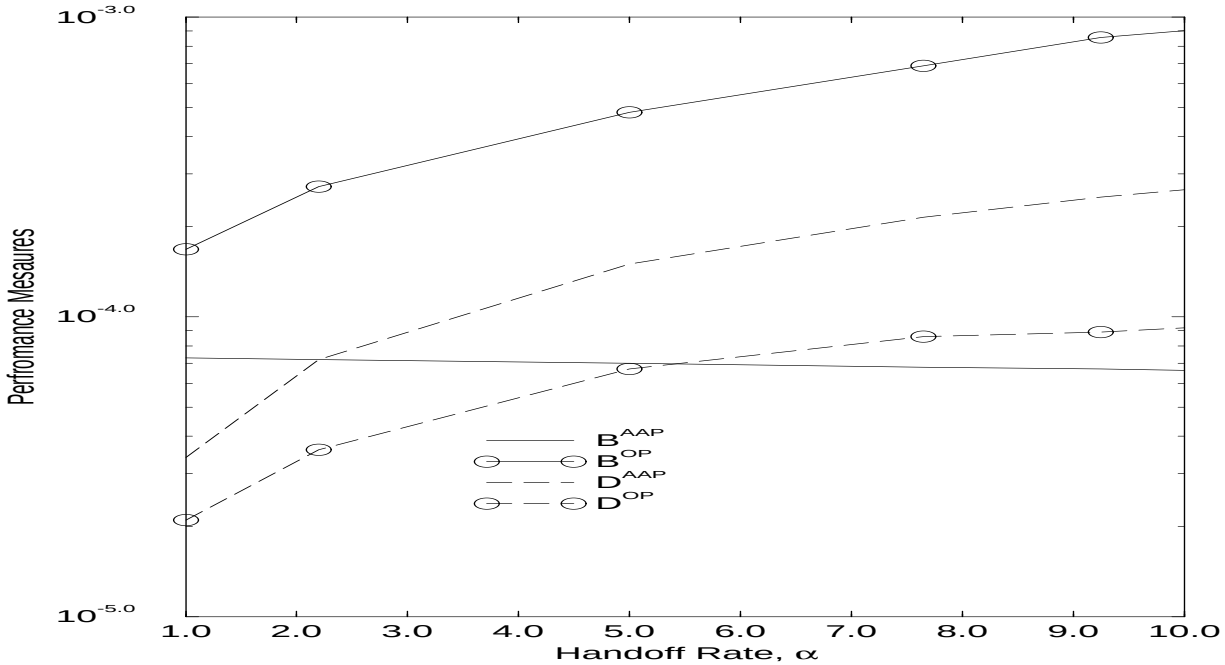

Figure 4.11: Performance evaluation of a ring system by optimizing weighted call incompletion rate under variable handoff rate with $n = 4$, $\lambda = 1.0$, $\mu = 1.0$, $\omega = 10.0$, $c = 10$.

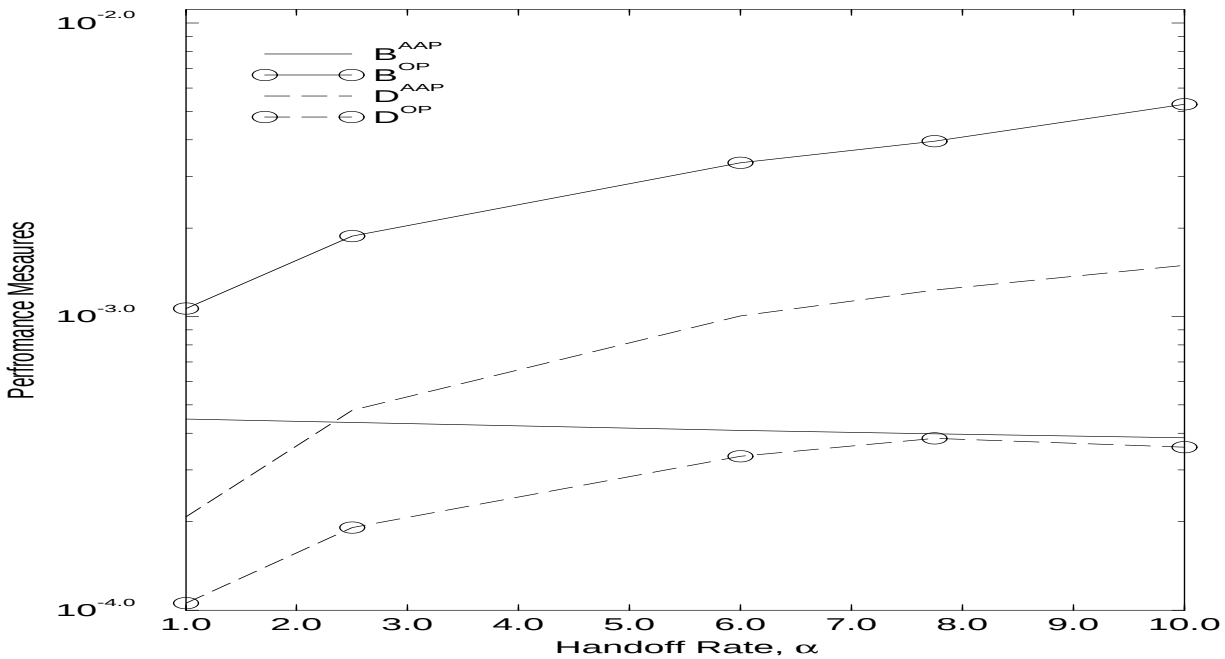

Figure 4.12: Performance evaluation of a ring system by optimizing weighted call incompletion rate under variable handoff rate with $n = 4$, $\lambda = 1.25$, $\mu = 1.0$, $\omega = 10.0$, $c = 10$.

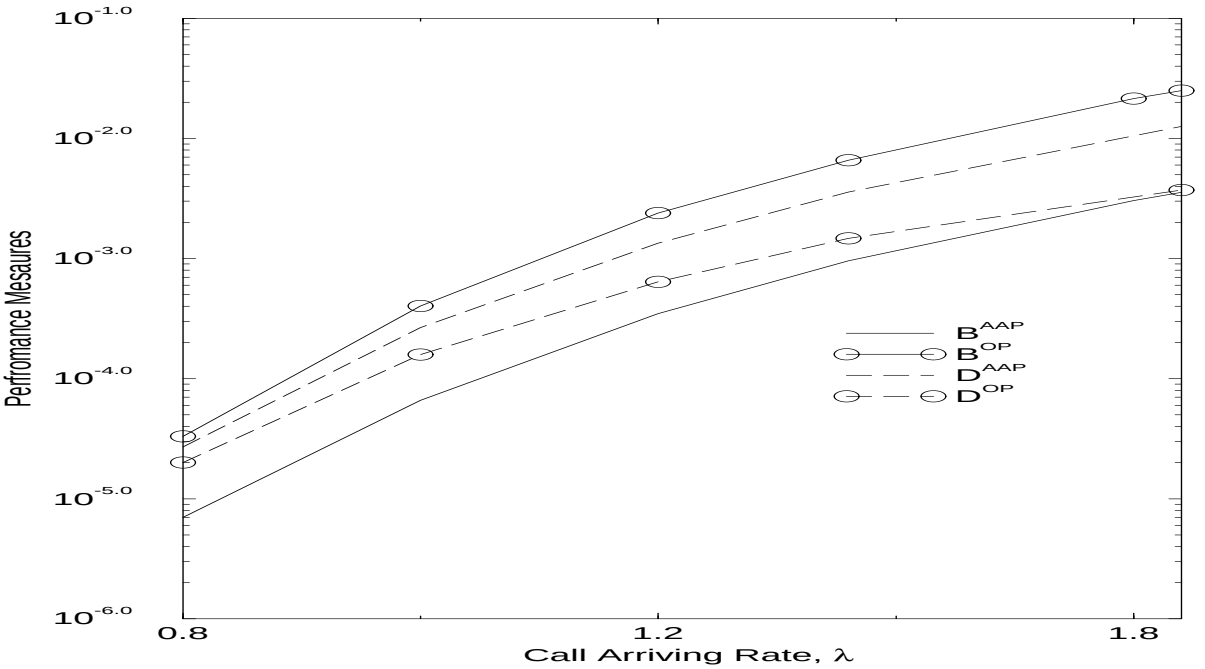

Figure 4.13: Performance evaluation of a ring system by optimizing weighted call incompletion rate under variable new call arriving rate with $n = 4$, $\mu = 1.0$, $\alpha = 10.0$, $\omega = 5.0$, $c = 10$.

where performances are measured as a function of handoff rate at two different loads of the system.

Suppose we have two different systems which differ only by the mobility of their calls and both systems have the same penalty for dropping a handoff over blocking a new call. In this situation, the new calls will experience lower blocking probability in the system with lower mobility. We can think these two systems as describing either a highway with a speed limit 55 mph or a local road with the speed limit of 25 mph.

4.3.3 Performance Measures with Different Load

If the weight, ω does not vary with the increase of offered load then the new call blocking and probability of handoff dropping both will increase; see Figure 4.13 and Figure 4.14. These are natural consequences of increasing offered load in the system.

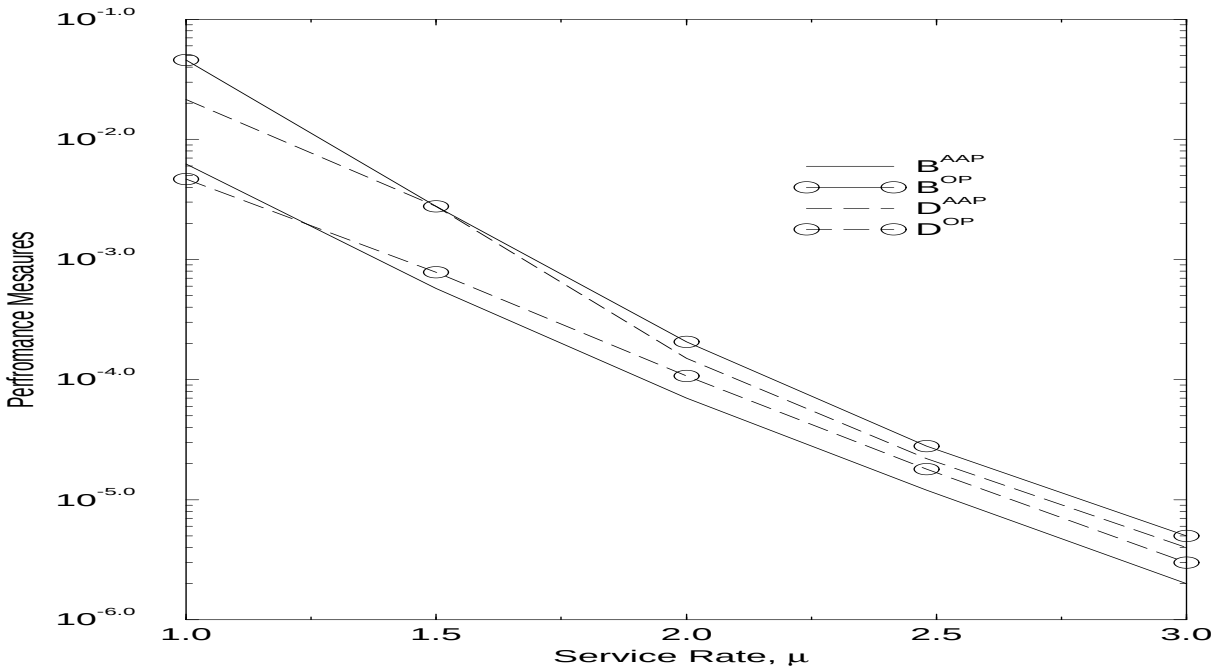

Figure 4.14: Performance evaluation of a ring system by optimizing weighted call incompletion rate under variable service rate with $n = 4$, $\lambda = 1.0$, $\alpha = 10.0$, $\omega = 5.0$, $c = 10$.

Chapter 5

Empirical Performance Measures for a Two-Class Cellular Network

5.1 Introduction

The cellular network can be used for many types of communications. Cellular phones can be used in cars as well as in the pockets of pedestrians. Nowadays a lot of attention is also paid to the integration of cellular network with the land wired communications networks for communicating not only voice messages but also data messages. This is more applicable to future systems as opposed to data over current voice system. We can consider a two-class system where pedestrians and customers in automobiles are two different classes or perhaps voice and data are two different classes. For example, pedestrians will have a much smaller handoff rate than the fast moving customers in cars, similarly, data messages can be distinguished from the voice messages by the shorter service requests.

In this chapter, under the minimization of weighted call incompletion rate, a comparison study is performed of a system with two classes of calls, those calls will differ in their relative arrival, service rate or handoff rate. In each experiment, two distinct and independent call classes (say, class 1 and class 2) are considered and the class-wise system performances are measured via the modified value-iteration algorithm for the optimal policy as well as for the admit all policy.

Experiments in this chapter are performed in two stages: first, maximization of call completion rate, and second, minimization of weighted call incompletion rate where $\omega > 1$.

5.1.1 Definitions of the Class-Wise System Parameters

Let λ_j denotes the call arriving rate (calls/sec) of class j calls at each cell, μ_j and α_j are their average service rate (calls/sec) and average handoff rate (calls/sec) respectively, where $j = 1, 2$. The number of cells in the system is denoted by n, and c represents the number of available channels. Normalized offered load from class j calls is ρ_j, where $\rho_j = (n\lambda_j)/(c\mu_j)$ and the relative service speed of the two classes calls is β, where $\beta = \mu_1/\mu_2$.

5.1.2 Performance Measures

Definitions of the Parameters for Class-Wise Performance Measures

Let cc_j^π denotes the average rate of call completion (calls/sec) for class j calls under the stationary policy π, where b_j^π and d_j^π are their average rate of new call blocking (calls/sec) and average rate of handoff dropping (calls/sec) respectively, see Figure 5.1.

Class-Wise Performance Measures

Using the equation 4.1 the probability of new call blocking for class j under the policy π is,

$$B_j^\pi = b_j^\pi/(n\lambda_j) \tag{5.1}$$

Similarly, the probability of handoff dropping for class j under the policy π is,

$$D_j^\pi = d_j^\pi/(n\lambda_j) \tag{5.2}$$

As a result, the probability of call completion for class j under the policy π is,

$$CC_j^\pi = 1 - B_j^\pi - D_j^\pi \tag{5.3}$$

Overall Performance Measures

Using the Bayes' rule in equation 5.1 and equation 5.2, we can write

The probability of new call blocking of the system under the policy π is

$$B^\pi = \frac{\lambda_1 B_1^\pi + \lambda_2 B_2^\pi}{\lambda_1 + \lambda_2} \tag{5.4}$$

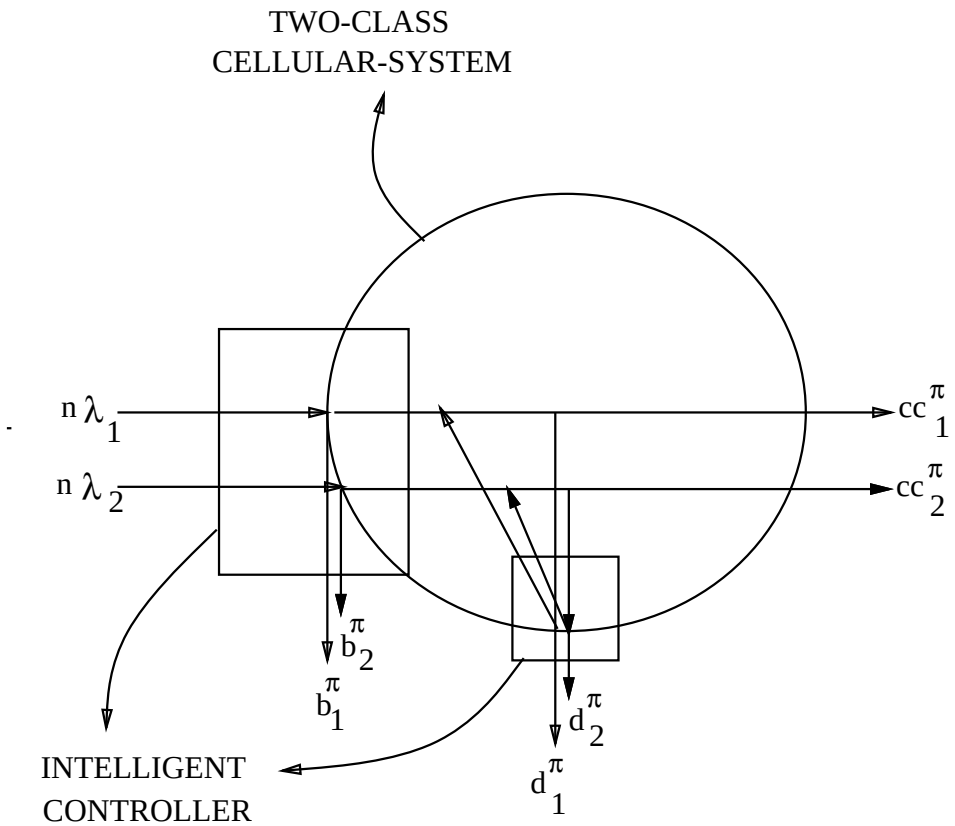

Figure 5.1: Class-wise average rate flow in a two-class mobile cellular system under the stationary policy π.

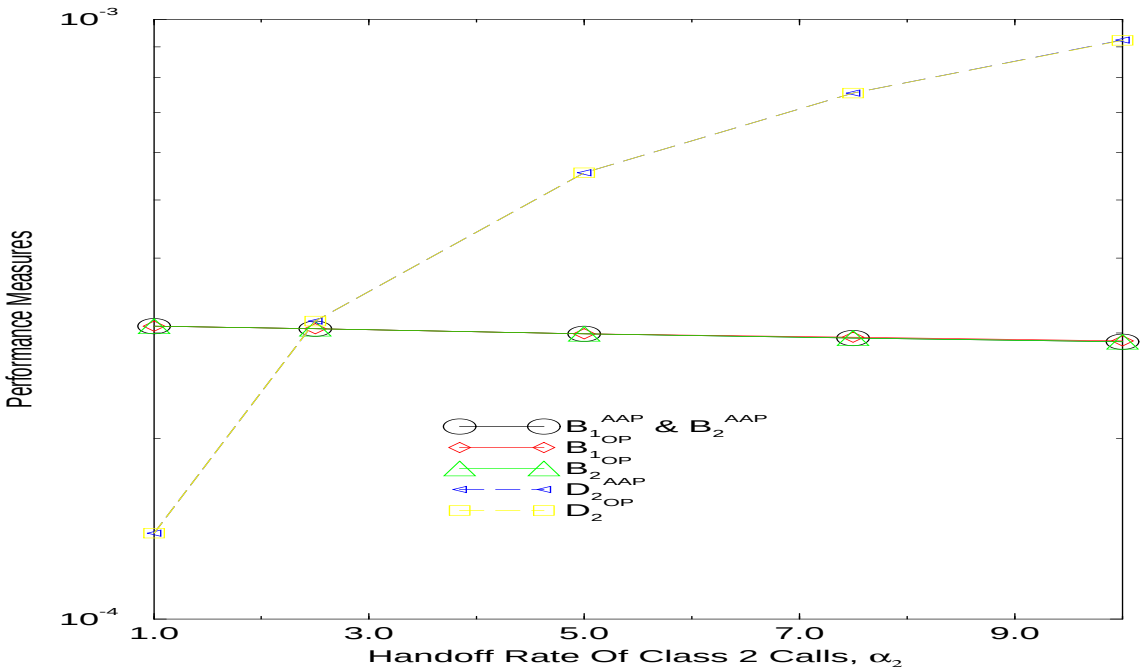

Figure 5.2: Performance evaluation of a two-class ring system by maximizing call completion rate under variable handoff rate of the class 2 calls with $n = 4$, $\lambda_1 = \lambda_2 = 0.2$, $\mu_1 = \mu_2 = 1.0$, $\alpha_1 = 0.0$, $c = 6$.

The probability of handoff dropping of the system under the policy π is

$$D^{\pi} = \frac{\lambda_1 D_1^{\pi} + \lambda_2 D_2^{\pi}}{\lambda_1 + \lambda_2} \tag{5.5}$$

and the probability of call completion of the system under the policy π is

$$CC^{\pi} = 1 - B^{\pi} - D^{\pi} \tag{5.6}$$

5.2 Maximizing Call Completion Rate

In Figure 5.2, class-wise performance measures are plotted as a function of handoff rate of class 2 calls. Here it is assumed that class 1 calls are stationary, $\alpha_1 = 0.0$, both classes have the same call arriving rate $\lambda_1 = \lambda_2 = 0.2$ and service rate, $\mu_1 = \mu_2 = 1.0$. In single class system, it has been observed that the optimal policy never differed significantly from the admit all policy if the handoff rate of the calls was varied. As a result high probability of handoff dropping was observed at higher handoff rate. Figure 5.2 also shows the same kind of results for two-class system.

In a two-class system, if one class is much faster in service than other (when $\beta \gg 1$ or $\beta \ll 1$), assigning a channel to a faster call system will be able to carry a another new

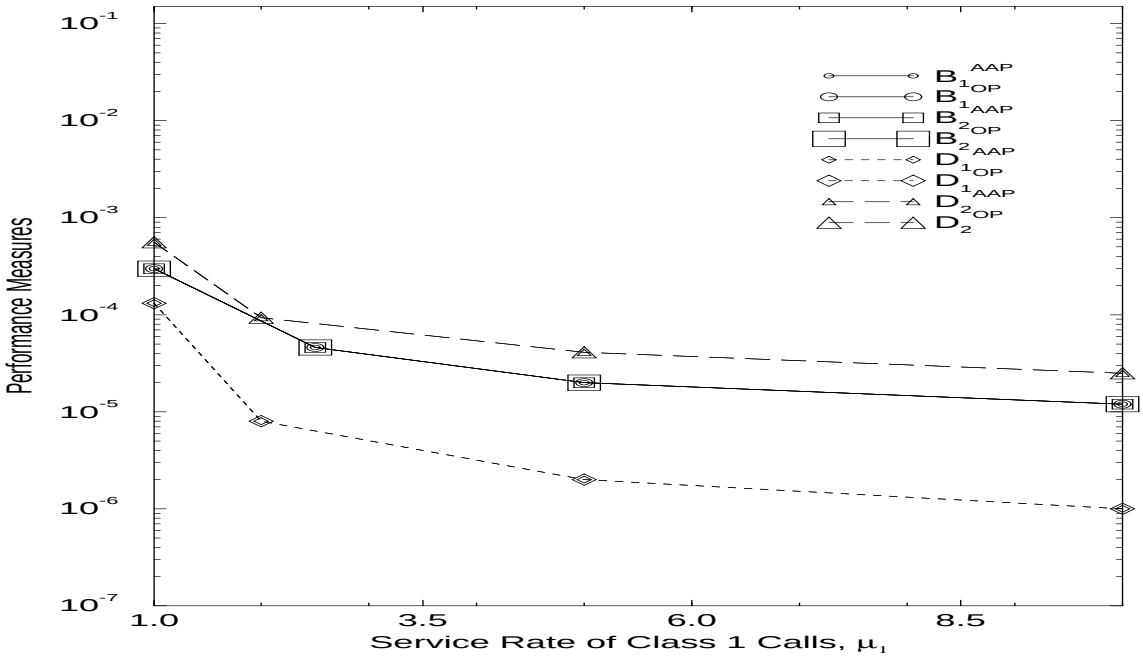

Figure 5.3: Performance evaluation of a two-class ring system by maximizing call completion rate under variable service rate of class 1 calls with $n = 4$, $\lambda_1 = \lambda_2 = 0.2$, $\mu_2 = 1.0$, $\alpha_1 = 1.0$, $\alpha_2 = 5.0$, $c = 6$.

call or a handoff sooner than a channel assigned to a slower call. As a result the slower call may experience higher blocking probability and system will achieve maximized call completion rate by serving more the faster class calls. An experiment is performed in Figure 5.3, where the service rate of class 1, μ_1 is varied with $\lambda_1 = \lambda_2 = 0.2$, $\mu_2 = 1.0$, $\alpha_1 = 0.0$, $\alpha_2 = 5.0$. Here μ_1 is made as large as ten times μ_2. But still no change in the system performance is observed.

For observing improvement in the overall call completion rate under the maximization of call completion rate, two separate experiments are performed. The experimental results are summarized in the Figure 5.4. These two experimental results imply that performance of maximizing call completion rate will be significant when CC^{AAP} has small value and $\beta \gg 1$. This kind of situation is excluded from this study. Because we want to see the effectiveness of maximizing call completion rate when there exists a reasonable probability of call completion (say, 0.97) under the admit all policy and how that effects the probability of handoff dropping of the mobile calls. But from Figure 5.3, it is observed that the maximization of call completion rate does not provide any kind of improvement in the desired operating range.

Maximizing Call Completion Rate with $\beta = 10.0$										
Observation	$\lambda_1 = \lambda_2$	μ_1	μ_2	α_1	α_2	CC^{AAP}	$B_1^{AAP} = B_2^{AAP}$	CC^{OP}	B_1^{OP}	B_2^{OP}
i	2.0	10.0	1.0	0.0	5.0	0.785	0.187	0.817	0.273	0.069
ii	4.0	10.0	1.0	0.0	5.0	0.488	0.460	0.628	0.645	0.0715

Figure 5.4: Experimental class-wise system parameters and performance measures for maximizing call completion rate with $n = 4$, $c = 6$.

5.2.1 Conclusion

In a two-class system, for a reasonable set of parameters we do not observe any improvement in the system performance under the maximization of call completion rate. In Figure 5.2 and Figure 5.3, it is also observed that the chance of dropping a conversation during the time of handoff increases with the increase of offered load and handoff rate. But low probability of handoff dropping is defined as the reliability of the service of the system. It is not desirable that reliability of the service will be lost due to increase of speed and load. Therefore, to keep the reliability of the cellular service, mobile calls should be provided with low probability of handoff dropping at a high speed and at a high load. We can handle this situation, as we did in the single class system, by minimizing the weighted call incompletion rate, where the value of the weight, ω is always greater than 1.

5.3 Optimization of Weighted Call Incompletion Rate

The study related to the optimization of weighted call incompletion rate will be performed in two stages. First, we try to find out analytically the comparative relation of class-wise performance measures of two classes calls when they will differ in their relative service rate and handoff rate. This analysis is performed with the help of the decomposed modified value-iteration algorithm and the basic definitions of the MDP parameters. Second, the comparative experimental results will be observed.

5.4 Analysis of the Comparative Relation of Class-Wise Performance Measures

For the notational advantages, here we shall choose an arbitrary linear system of n cells. This analysis can be extended to any size of the ring system, or to any structure of the system.

Here we shall extend some of the notations of the system parameters used in the Chapter 3. The current state of the system will be represented by a vector, $x = (x_{11}, x_{12}, \ldots, x_{n1}, x_{n2})$, where x_{ij} is the number of class j calls in cell i, where $j = 1, 2$. In each state x, the decision variables of class j calls in the cell i will be $a_{ij}(x)$, $h_{(i-1,i)j}(x)$ and $h_{(i+1,i)j}(x)$, where

$$a_{(i)j}(x) = \begin{cases} 1 & \text{if a new class } j \text{ call is accepted at cell } i. \\ 0 & \text{if a new class } j \text{ call is blocked at cell } i. \end{cases}$$

$$h_{(i-1,i)j}(x) = \begin{cases} 1 & \text{if a class } j \text{ handoff from cell } i-1 \text{ to cell } i \text{ is accepted.} \\ 0 & \text{if a class } j \text{ handoff from cell } i-1 \text{ to cell } i \text{ is blocked.} \end{cases}$$

$$h_{(i+1,i)j}(x) = \begin{cases} 1 & \text{if a class } j \text{ handoff from cell } i+1 \text{ to cell } i \text{ is accepted.} \\ 0 & \text{if a class } j \text{ handoff from cell } i+1 \text{ to cell } i \text{ is blocked.} \end{cases}$$

If there is any violation of any of the re-use constraints, the system will not provide a new channel to the new call or handoff, which means that call will leave the system immediately, see Chapter 3. Our goal is to minimize weighted call incompletion rate, where ($\omega > 1$). So the reward rate is

$$\begin{aligned} r(x,a) &= \sum_{j=1}^{2}\{\sum_{i=1}^{n}[-x_{ij}\omega\alpha_j(1-p_{(i)j})\overline{h_{(i,i-1)j}(x)} - \lambda_j\overline{a_{(i)j}(x)} - x_{ij}\omega\alpha_j p_{(i)j}\overline{h_{(i,i+1)j}(x)}]\} \\ &= -\sum_{j=1}^{2}\{\sum_{i=1}^{n}(\lambda_j + x_{ij}w\alpha_j)\} + \sum_{j=1}^{2}\{\sum_{i=1}^{n}[(x_{ij}\omega\alpha_j(1-p_{(i)j})h_{(i,i-1)j}(x) \\ &\quad + \lambda_j a_{(i)j}(x) + x_{ij}\omega\alpha_j p_{(i)j}h_{(i,i+1)j}(x)]\} \end{aligned} \tag{5.7}$$

where $p_{(i)j}$ equals the probability that a class j call in cell i requesting a handoff goes to cell $(i+1)$, and $(1-p_{(i)j})$ equals the probability that a class j call in cell i requesting a handoff goes to cell $(i-1)$.

The equation 5.7 can be assumed as the extended form of the equation 3.8 for the two-class system. So equations 3.9 and 3.10 can be written for class j calls as

$$a_{(i)j}(x) = 1[V_{m-1}(x + e_{(i)j}) + \Gamma > V_{m-1}(x)] \tag{5.8}$$

$$h_{(i,i-1)j}(x) = 1[V_{m-1}(x + e_{(i-1)j} - e_{(i)j}) + \omega\Gamma > V_{m-1}(x - e_{(i)j})] \tag{5.9}$$

where $e_{(i)j}$ is a vector of zeroes, except for a one for the $\{2(i-1)+j\}$th component.

In Chapter 3, $V_m(x)$ was defined as the maximum total expected costs with m periods left to the time horizon when the current state is x. From this definition, we can write,

$$\lim_{m\to\infty} \frac{V_m(x)}{m} = g^* \tag{5.10}$$

where g^* is the expected costs of the system under the optimal policy. Equation 5.10 finds the existence of the values $v(x)$, $x \in \Lambda$, such that, for each $x \in \Lambda$,

$$V_m(x) \approx mg^* + v(x) \qquad \text{for large } m. \tag{5.11}$$

where $v(x)$ is defined as the relative value of the state x. For large m, we can also write,

$$v(x) - v(y) \approx V_{m-1}(x) - V_{m-1}(y) \tag{5.12}$$

where $v(x) - v(y)$ measures the difference in total expected costs over an infinitely long period of time starting in state x rather than in state y when the system is ruled by the optimal policy, see p.166; [Tij86].

Slow and Fast Moving Customers

Consider the case in which $\mu_1 = \mu_2$ but $\alpha_1 < \alpha_2$. Here the class 1 calls are moving slowly and thus have relatively low handoff rate. If there is a question of choice between the state, $x + e_{(i)1}$ and $x + e_{(i)2}$ to start the system in order to achieve the minimum weighted handoff call incompletion rate in the long run, when $\omega > 1$. It is likely that the state, $x + e_{(i)1}$ will be selected. This selection basically depends on the relative choice between having a class 1 call and a class 2 call, because all the other customers are the same in those two states. A call of either of both classes requires the same amount of time in the system on average ($1/\mu_1 = 1/\mu_2$), so to complete a full conversation a

class 1 call will make fewer number of handoffs on average. A successfully completed class 1 call makes $\frac{\mu_1+\alpha_1}{\mu_1}$ number of handoffs on average, where a class 2 call makes $\frac{\mu_2+\alpha_2}{\mu_2}$ number of handoffs. As a class 1 call making fewer handoffs on average, it is less likely to be dropped during the time of its conversation. This also implies if the system starts from $x + e_{(i)1}$ then it has lesser chance to pay higher penalty in the long run than starting from the state, $x + e_{(i)2}$. That is, if the system starts from the state $x + e_{(i)1}$ then total long run costs will be higher than if it starts from the state, $x + e_{(i)2}$. From the definition of the relative value of the state, $v(x + e_{(i)1}) - v(x + e_{(i)2}) > 0$, or equivalently, for large m, equation 5.12 implies

$$V_{m-1}(x + e_{(i)1}) > V_{m-1}(x + e_{(i)2}) \tag{5.13}$$

If we re-write the equation 5.8 for each class then,

$$a_{(i)1}(x) = 1[V_{m-1}(x + e_{(i)1}) + \Gamma > V_{m-1}(x)] \tag{5.14}$$

$$a_{(i)2}(x) = 1[V_{m-1}(x + e_{(i)2}) + \Gamma > V_{m-1}(x)] \tag{5.15}$$

Comparing equation 5.14 and equation 5.15, the equation 5.13 implies that if we shall accept a class 2 call in any cell at any state, then we shall accept a class 1 call in that cell at that state definitely. However, the converse does not follow. That will make the set of blocking states for class 1 calls, Ψ_1 a subset of that of class 2 calls, Ψ_2, $\Psi_1 \subset \Psi_2$. As a result a new class 2 call will experience higher probability of blocking to enter the system, $B_1^{OP} < B_2^{OP}$. As expected, this result is also observed in our experiments.

For the state u and r, where $u = x + e_{(i-1)1}$, $r = x + e_{(i-1)2}$, the optimal policy $h_{(i-1,i)1}(u)$ and $h_{(i-1,i)2}(r)$ will be (from the equation 5.9)

$$\begin{aligned} h_{(i-1,i)1}(u) &= 1[V_{m-1}(u + e_{(i)1} - e_{(i-1)1}) + \omega\Gamma > V_{m-1}(u - e_{(i-1)1})] \\ &= 1[V_{m-1}(x + e_{(i)1}) + \omega\Gamma > V_{m-1}(x)] \end{aligned} \tag{5.16}$$

$$\begin{aligned} h_{(i-1,i)1}(r) &= 1[V_{m-1}(r + e_{(i)2} - e_{(i-1)2}) + \omega\Gamma > V_{m-1}(r - e_{(i-1)2})] \\ &= 1[V_{m-1}(x + e_{(i)2}) + \omega\Gamma > V_{m-1}(x)] \end{aligned} \tag{5.17}$$

Comparing equation 5.16 and equation 5.17, the equation 5.13 implies that if we shall accept a class 2 handoff from cell $i-1$ to cell i at state r, then we shall accept

a class 1 handoff from cell $i-1$ to cell i at state u definitely. The converse is not necessarily true. As a result the set of the states, which are entered by blocking class 1 handoffs, will be a subset of the set of the states, which are entered by blocking class 2 handoffs. We assume, $\alpha_1 < \alpha_2$, so it is reasonable to expect that the average rate of handoff dropping of class 1 calls will be less than that of class 2 calls. That means, $\lambda_1 D_1^{OP} < \lambda_2 D_2^{OP}$ or $D_2^{OP} > \frac{\lambda_1}{\lambda_2} D_1^{OP}$. So when $\lambda_1 \geq \lambda_2$, then it reasonable to observe $D_2^{OP} > D_1^{OP}$. But for $D_2^{OP} < D_1^{OP}$, λ_2 needs to be greater than λ_1. In Figure 5.16 we see that for fixed λ_1, D_1^{OP} and D_2^{OP} increase with the increase in offered load due to increasing λ_2 and when $\lambda_2 > \hat{\lambda} > \lambda_1$, where $\hat{\lambda} \approx 0.625$, it is observed that $D_2^{OP} < D_1^{OP}$.

Short Class 1 Calls

Here both classes have the same mobility. But due to the higher service rate of class 1 calls, if the system starts from the state, $x + e_{(i)1}$, then it will probably be able to serve a handoff sooner than if it starts from the state $x + e_{(i)2}$. Again, a class 1 call makes fewer number of handoffs on average to finish a successful conversation. So, as analyzed previously, it is reasonable to expect higher long run costs will be achieved, if the system starts from the state, $x + e_{(i)1}$ rather than the state, $x + e_{(i)2}$. So, we can equate, $V_{m-1}(x + e_{(i)1}) > V_{m-1}(x + e_{(i)2})$. We have seen that this relation provides $B_1^{OP} < B_2^{OP}$, for all values of λ_1, λ_2, and $D_1^{OP} < D_2^{OP}$, when $\lambda_1 \geq \lambda_2$. But for $D_1^{OP} > D_2^{OP}$, λ_2 needs to be greater than λ_1.

Combining the above analysises, for $\mu_1 > \mu_2$ and $\alpha_1 < \alpha_2$, we can also expect, $B_1^{OP} < B_2^{OP}$, for all values of λ_1, λ_2, and $D_1^{OP} < D_2^{OP}$, when $\lambda_1 \geq \lambda_2$. But for $D_1^{OP} > D_2^{OP}$, λ_2 needs to be greater than λ_1.

From our previous discussions the situation, $\mu_1 > \mu_2$ and $\alpha_1 > \alpha_2$ was excluded. We have observed that for minimizing the weighted call incompletion rate, the system prefers to serve those calls which have shorter service requests and smaller handoff rate. Here class 1 calls posses shorter service requests but higher handoff rate. So, the stochastic properties of class 1 calls fulfill the partial requirements to get preferential service over class 2 calls under minimization of weighted incompletion rate. For this situation, to observe the comparative relation of class-wise performance measures an

Variable Weighting Factor						
$\lambda_1 = \lambda_2$	$\mu_1 = \mu_2$	α_1	α_2	ω	Observation	Comparative relation of the optimal class-wise performance measures
0.2	1.0	0.0	10.0	variable $1 \leq \omega \leq 10$	Figure 5.9	$B_1^{OP} < B_2^{OP}$ $D_1^{OP} < D_2^{OP}$
0.4	1.0	0.0	10.0	variable $1 \leq \omega \leq 10$	Figure 5.10	$B_1^{OP} < B_2^{OP}$ $D_1^{OP} < D_2^{OP}$

Figure 5.5: Experimental class-wise system parameters and comparative relation of the class-wise optimal performance measures under variable weighting factor with $n = 4$, $c = 6$.

experiment is performed in Figure 5.20. Here the service rate of class 1 calls is assumed as a variable, when the other parameters are, $\lambda_1 = \lambda_2 = 0.2$, $\mu_2 = 1.0$, $\alpha_1 = 5.0$, $\alpha_2 = 1.0$, $\omega = 10.0$. As the value of μ_1 decreases the overall load of the system also decreases. As a result the new calls of both classes are experiencing lower probability of blocking with the decrease of μ_1. With the decrease of the B_1^{OP}, the class 1 calls will get into the system more frequently. As those calls have higher mobility, the system likes to keep $D_1^{OP} < D_2^{OP}$ after certain value of μ_1 and that will save the system from paying a higher amount of penalty. Of course the system will prefer to block class 1 new calls than to drop class 1 handoffs. As a result, class 1 calls will experience higher blocking probability, $B_1^{OP} > B_2^{OP}$.

5.4.1 Comparative Relation Of The Class-Wise Performance Measures

The comparative results of class-wise performance measures under different system variables are summarized in the following tables.

Different Mobility					
$\lambda_1 = \lambda_2$	$\mu_1 = \mu_2$	α_1	α_2	Observation	Comparative relation of the optimal class-wise performance measures
0.2	1.0	0.0	variable $1 \leq \alpha_2 \leq 10$	Figure 5.11	$B_1^{OP} < B_2^{OP}$ $D_1^{OP} < D_2^{OP}$
0.4	1.0	0.0	variable $1 \leq \alpha_2 \leq 10$	Figure 5.12	$B_1^{OP} < B_2^{OP}$ $D_1^{OP} < D_2^{OP}$
0.2	1.0	variable $1 \leq \alpha_1 \leq 5.0$	5.0	Figure 5.13	$B_1^{OP} < B_2^{OP}$ $D_1^{OP} < D_2^{OP}$

Figure 5.6: Experimental class-wise system parameters and comparative relation of the class-wise optimal performance measures under different mobility of the calls with $n = 4$, $\omega = 10.0$, $c = 6$.

Different Arriving Rate						
λ_1	λ_2	$\mu_1 = \mu_2$	α_1	α_2	Observation	Comparative relation of the optimal class-wise performance measures
variable $0.2 \leq \lambda_1 \leq 0.8$	0.2	1.0	1.0	10.0	Figure 5.14	$B_1^{OP} < B_2^{OP}$ $D_1^{OP} < D_2^{OP}$
0.2	variable $0.2 \leq \lambda_2 \leq 0.8$	1.0	0.0	10.0	Figure 5.15	$B_1^{OP} < B_2^{OP}$ $D_1^{OP} < D_2^{OP}$
0.2	variable $0.2 \leq \lambda_2 \leq 0.8$	1.0	1.0	10.0	Figure 5.16	$B_1^{OP} < B_2^{OP}$ $D_1^{OP} < D_2^{OP}; \lambda_2 < \hat{\lambda}$ $D_1^{OP} = D_2^{OP}; \lambda_2 = \hat{\lambda}$ $D_1^{OP} > D_2^{OP}; \lambda_2 > \hat{\lambda}$ $\hat{\lambda} \approx 0.625$

Figure 5.7: Experimental class-wise system parameters and comparative relation of the class-wise performance measures under different arriving rate with $n = 4$, $\omega = 10.0$, $c = 6$.

Different Service Rate							
λ_1	λ_2	μ_1	μ_2	α_1	α_2	Observation	Comparative relation of the optimal class-wise performance measures
0.1	0.2	variable $1.0 \leq \mu_1 \leq 10.0$	1.0	0.0	10.0	Figure 5.17	$B_1^{OP} < B_2^{OP}$ $D_1^{OP} < D_2^{OP}$
0.2	0.2	variable $1.0 \leq \mu_1 \leq 5.0$	1.0	5.0	5.0	Figure 5.18	$B_1^{OP} < B_2^{OP}$ $D_1^{OP} < D_2^{OP}$
0.2	0.2	variable $1.0 \leq \mu_1 \leq 5.0$	1.0	1.0	5.0	Figure 5.19	$B_1^{OP} < B_2^{OP}$ $D_1^{OP} < D_2^{OP}$
0.2	0.2	variable $1.0 \leq \mu_1 \leq 5.0$	1.0	5.0	1.0	Figure 5.20	$B_1^{OP} > B_2^{OP}$ $D_1^{OP} < D_2^{OP}; \mu_1 < \hat{\mu}$ $D_1^{OP} = D_2^{OP}; \mu_1 = \hat{\mu}$ $D_1^{OP} > D_2^{OP}; \mu_1 > \hat{\mu}$ where $\hat{\mu} \approx 1.8$

Figure 5.8: Experimental class-wise system parameters and comparative relation of the class-wise performance measures under different service rate with $n = 4$, $\omega = 10.0$, $c = 6$.

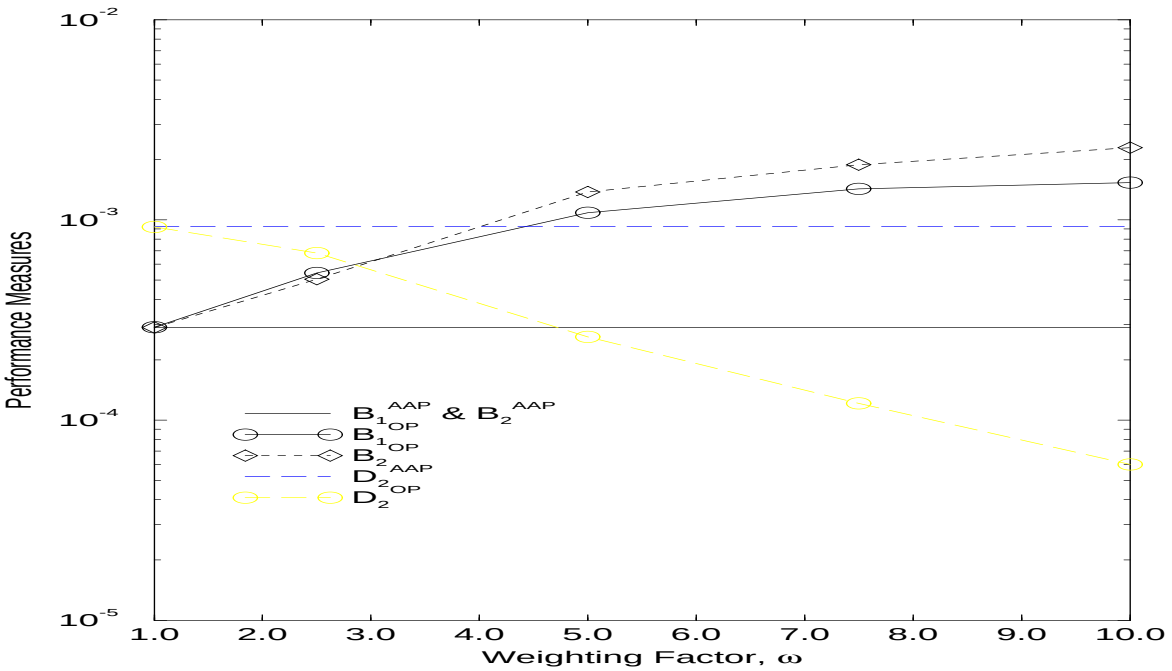

Figure 5.9: Performance evaluation of a two-class ring system by optimizing weighted call incompletion rate under variable weighting factor with $n = 4$, $\lambda_1 = \lambda_2 = 0.2$, $\mu_1 = \mu_2 = 1.0$, $\alpha_1 = 0.0$, $\alpha_2 = 10.0$, $c = 6$.

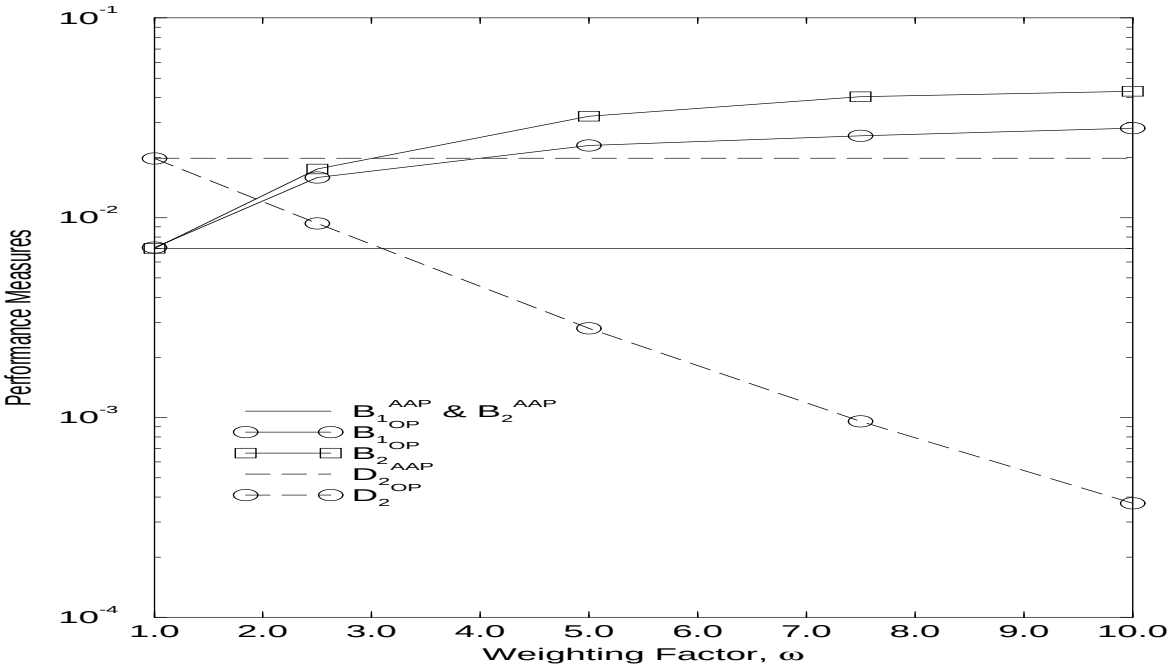

Figure 5.10: Performance evaluation of a two-class ring system by optimizing weighted call incompletion rate under variable weighting factor with $n = 4$, $\lambda_1 = \lambda_2 = 0.4$, $\mu_1 = \mu_2 = 1.0$, $\alpha_1 = 0.0$, $\alpha_2 = 10.0$, $c = 6$.

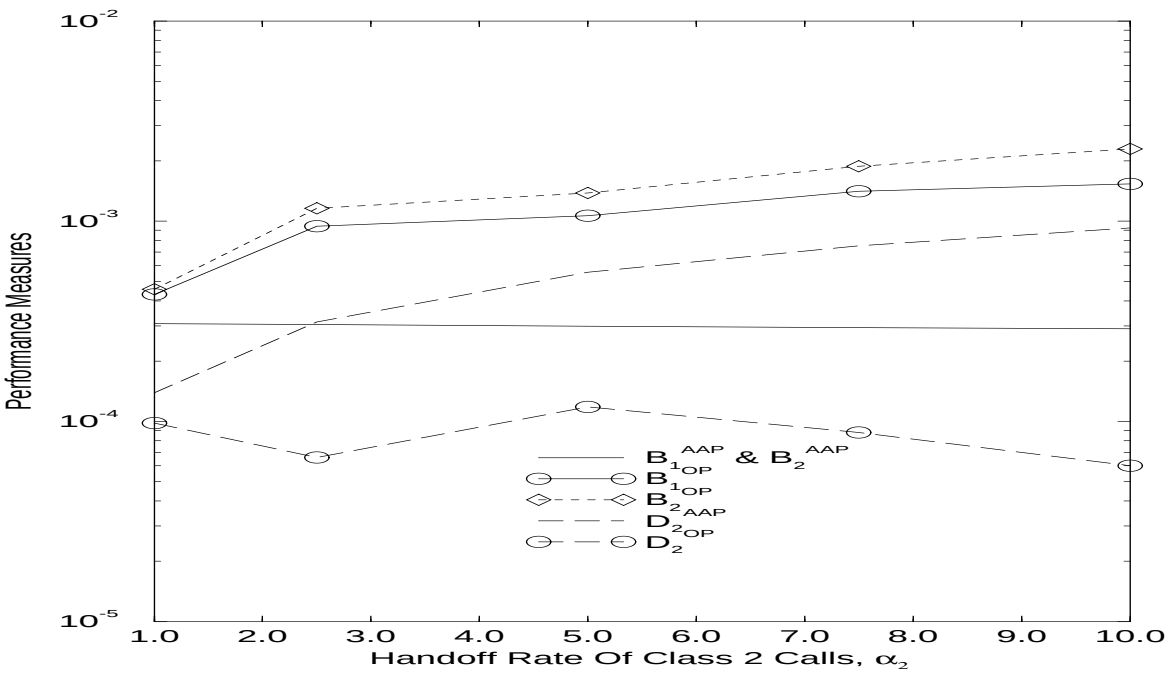

Figure 5.11: Performance evaluation of a two-class ring system by optimizing weighted call incompletion rate under different mobility of class 2 calls with $n = 4$, $\lambda_1 = \lambda_2 = 0.2$, $\mu_1 = \mu_2 = 1.0$, $\alpha_1 = 0.0$, $\omega = 10.0$, $c = 6$.

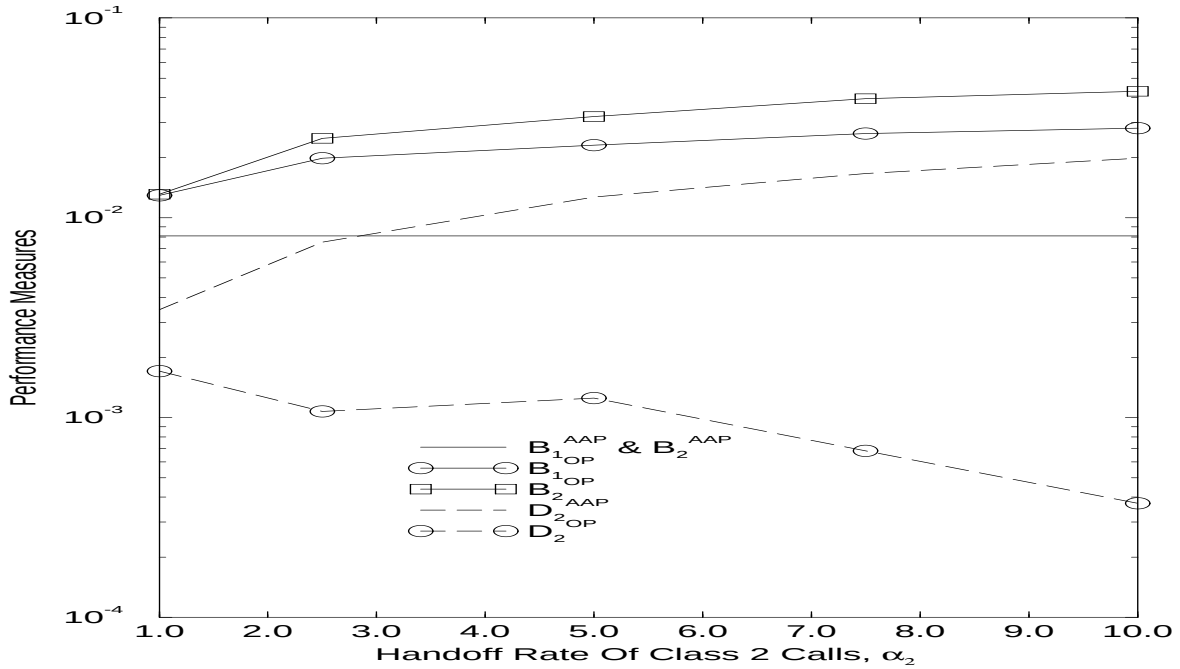

Figure 5.12: Performance evaluation of a two-class ring system by optimizing weighted call incompletion rate under different mobility of class 2 calls with $n = 4$, $\lambda_1 = \lambda_2 = 0.4$, $\mu_1 = \mu_2 = 1.0$, $\alpha_1 = 0.0$, $\omega = 10.0$, $c = 6$.

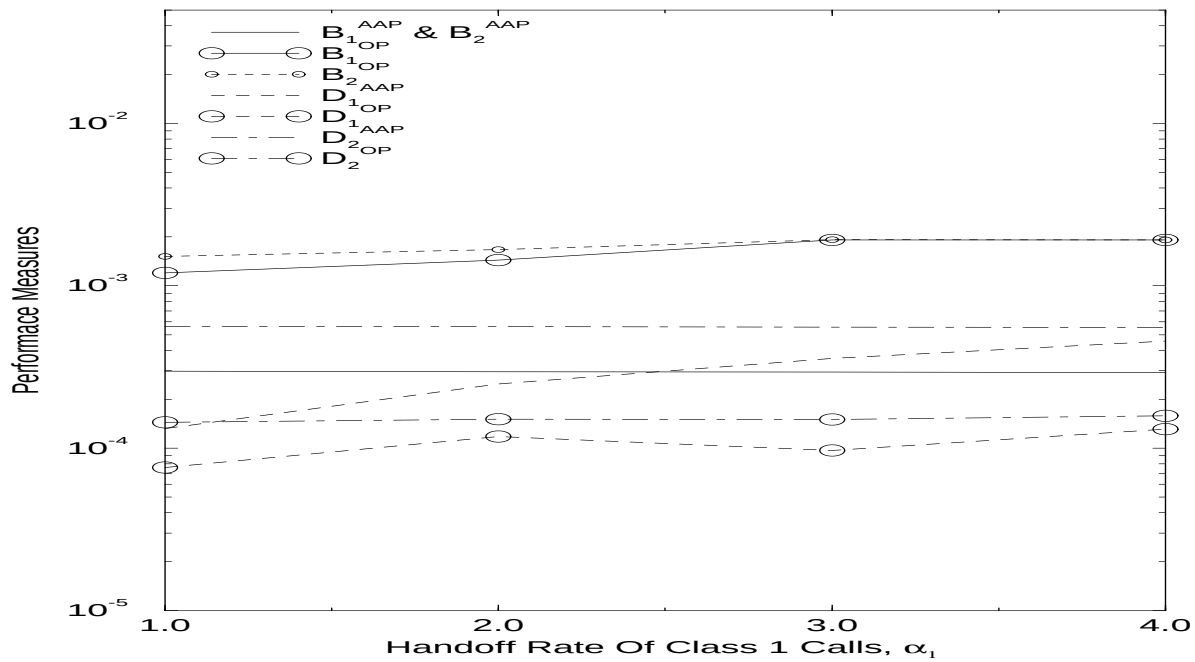

Figure 5.13: Performance evaluation of a two-class ring system by optimizing weighted call incompletion rate under different mobility of class 1 calls with $n = 4$, $\lambda_1 = \lambda_2 = 0.2$, $\mu_1 = \mu_2 = 1.0$, $\alpha_2 = 5.0$, $\omega = 10.0$, $c = 6$.

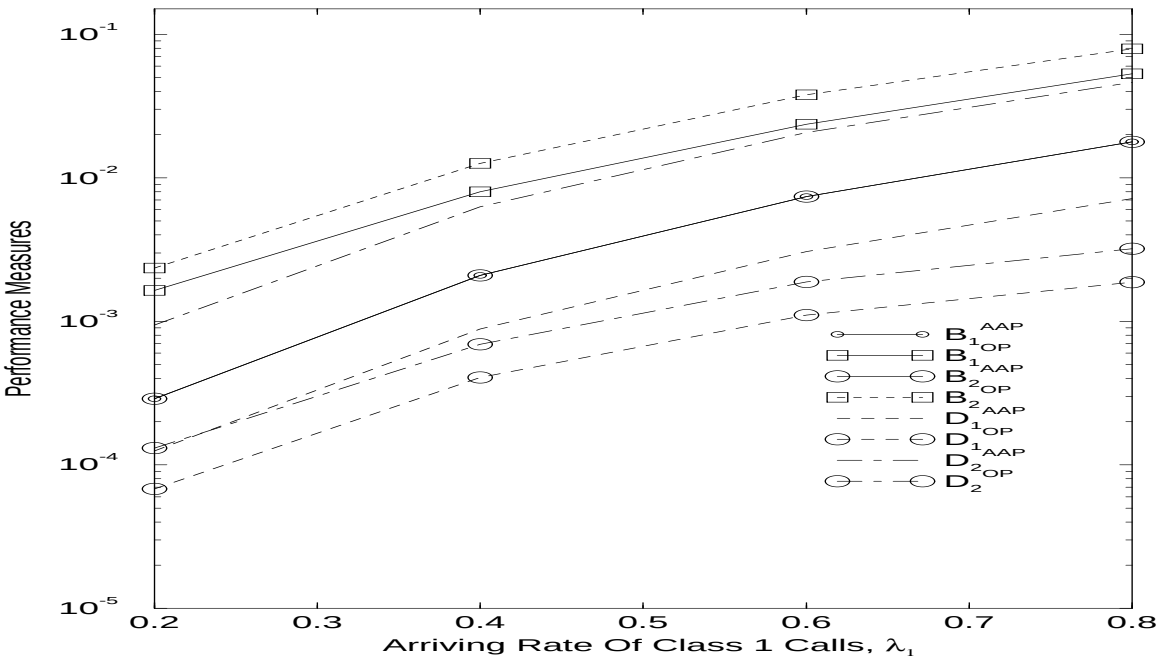

Figure 5.14: Performance evaluation of a two-class ring system by optimizing weighted call incompletion rate under variable arriving rate of class 1 calls with $n = 4$, $\lambda_2 = 0.2$, $\mu_1 = \mu_2 = 1.0$, $\alpha_1 = 1.0$, $\alpha_2 = 10.0$, $\omega = 10.0$, $c = 6$.

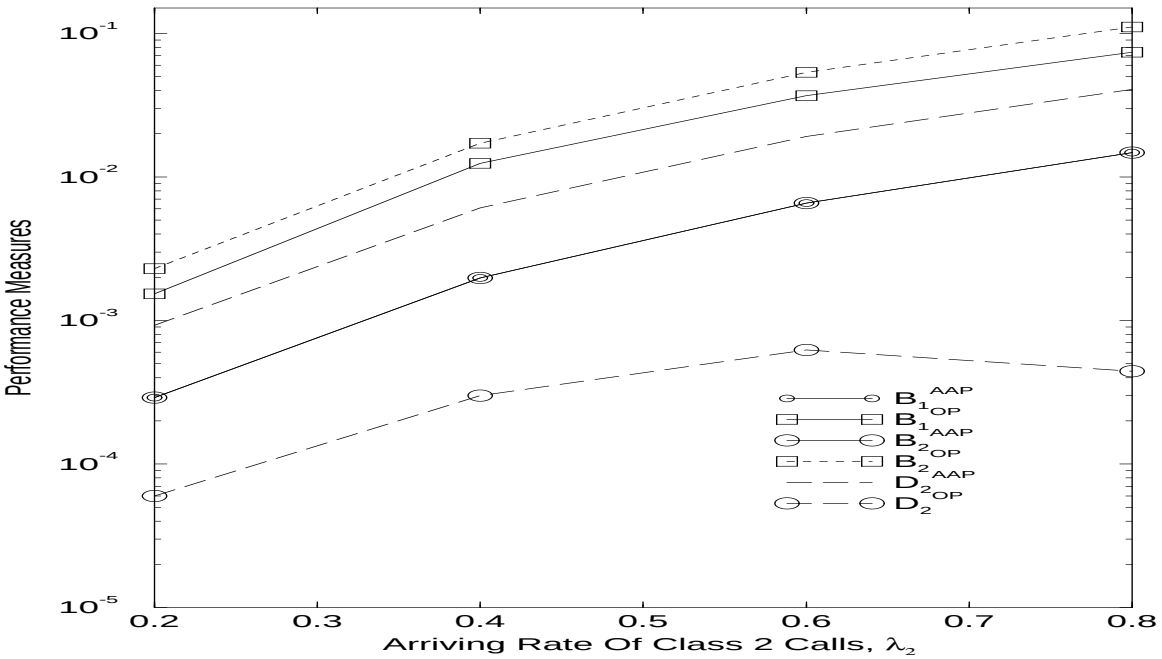

Figure 5.15: Performance evaluation of a two-class ring system by optimizing weighted call incompletion rate under variable arriving rate of class 2 calls with $n = 4$, $\lambda_1 = 0.2$, $\mu_1 = \mu_2 = 1.0$, $\alpha_1 = 0.0$, $\alpha_2 = 10.0$, $\omega = 10.0$, $c = 6$.

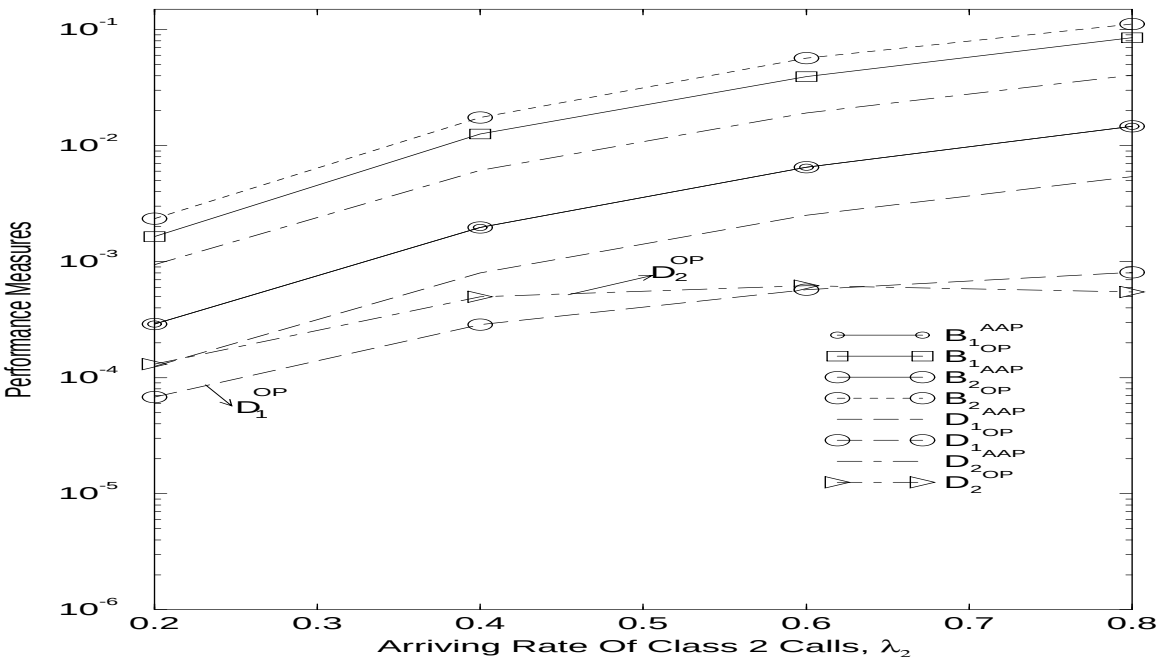

Figure 5.16: Performance evaluation of a two-class ring system by optimizing weighted call incompletion rate under variable arriving rate of class 2 calls with $n = 4$, $\lambda_1 = 0.2$, $\mu_1 = \mu_2 = 1.0$, $\alpha_1 = 1.0$, $\alpha_2 = 10.0$, $\omega = 10.0$, $c = 6$.

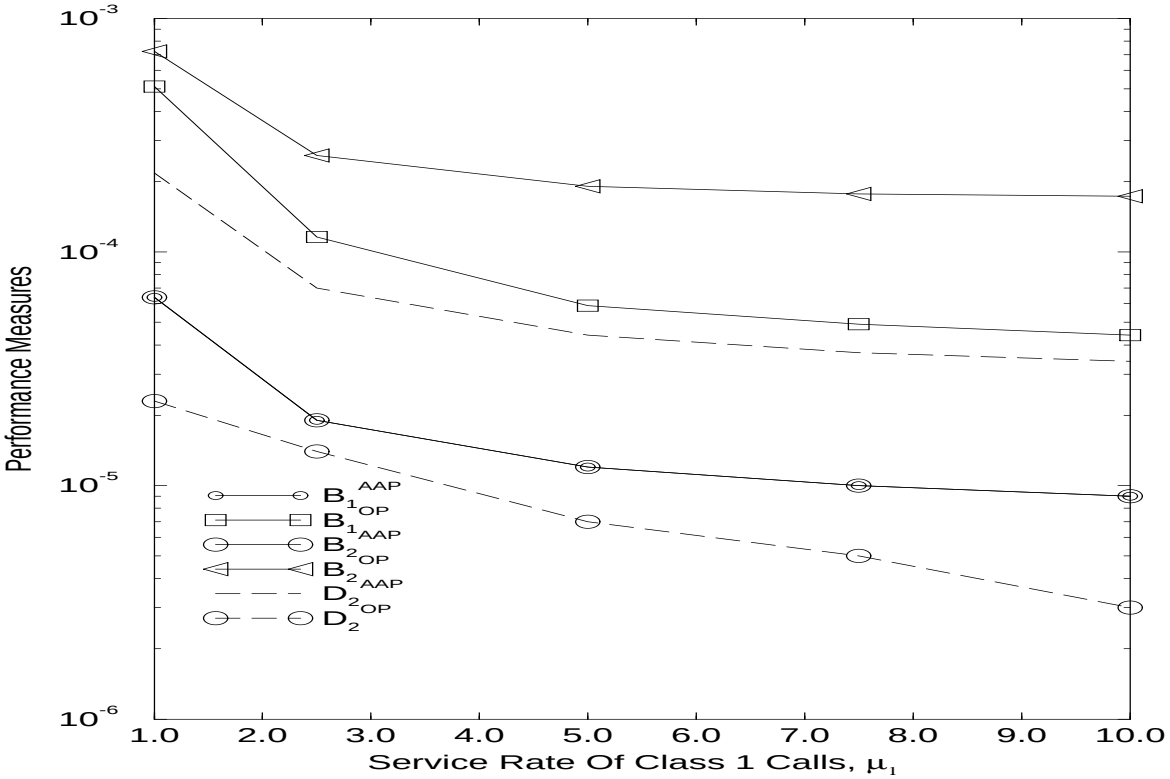

Figure 5.17: Performance evaluation of a two-class ring system by optimizing weighted call incompletion rate under variable service rate of class 1 calls with $n = 4$, $\lambda_1 = 0.1$, $\lambda_2 = 0.2$, $\mu_2 = 1.0$, $\alpha_1 = 0.0$, $\alpha_2 = 10.0$, $\omega = 10.0$, $c = 6$.

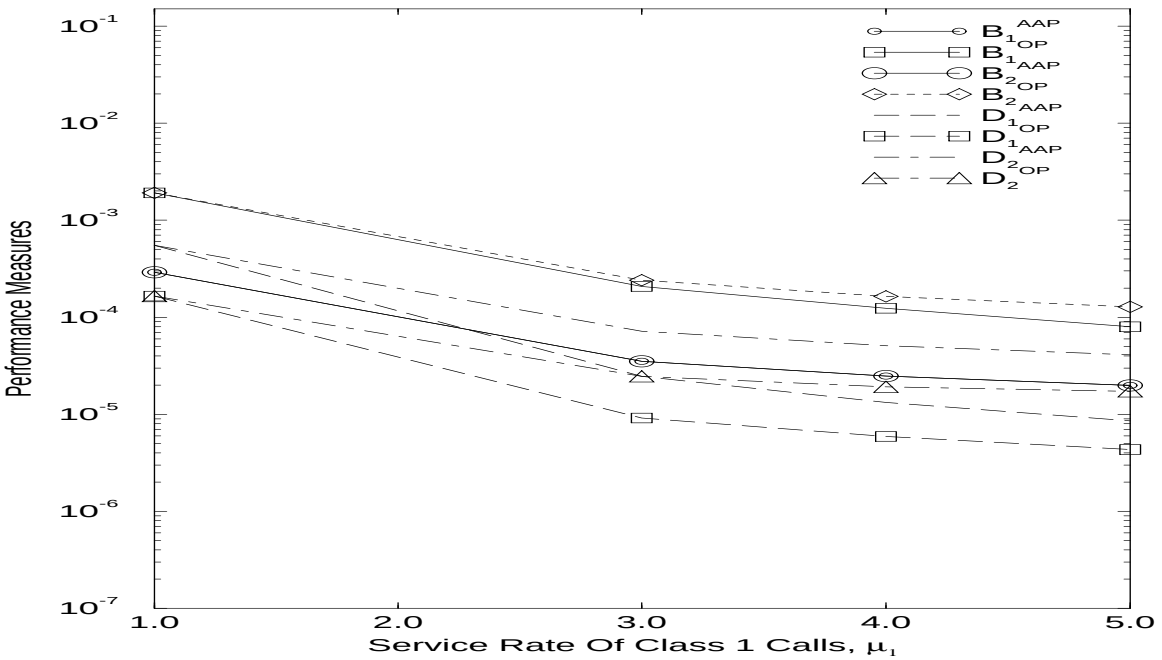

Figure 5.18: Performance evaluation of a two-class ring system by optimizing weighted call incompletion rate under variable service rate of class 1 calls with $n = 4$, $\lambda_1 = 0.2$, $\lambda_2 = 0.2$, $\mu_2 = 1.0$, $\alpha_1 = 5.0$, $\alpha_2 = 5.0$, $\omega = 10.0$, $c = 6$.

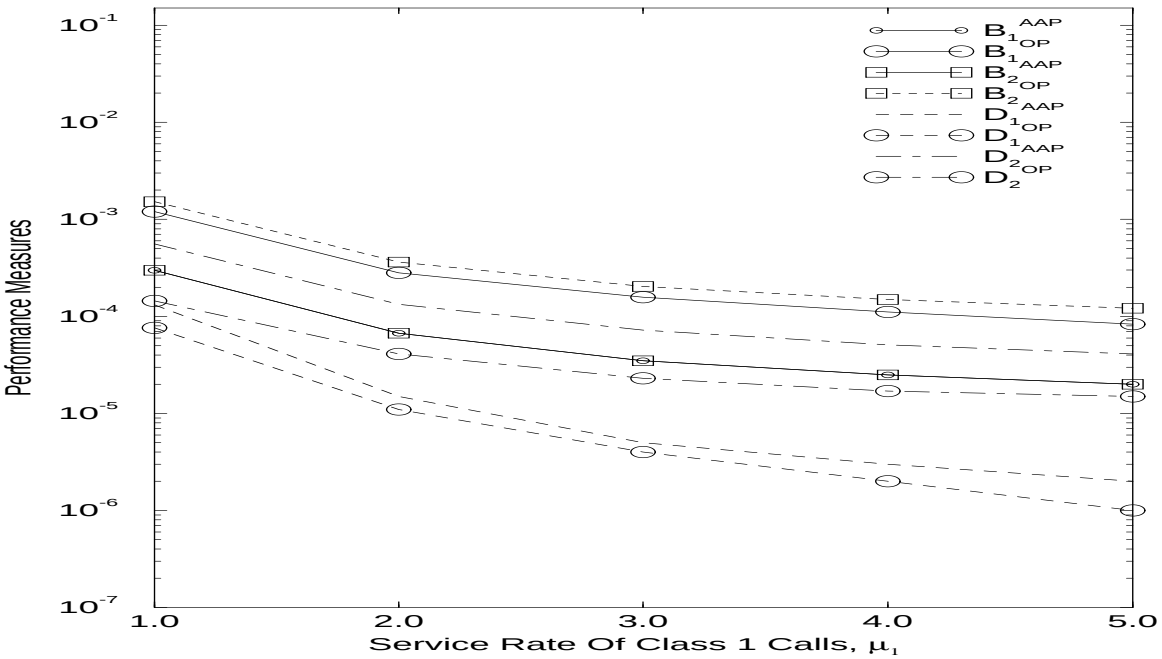

Figure 5.19: Performance evaluation of a two-class ring system by optimizing weighted call incompletion rate under variable service rate of class 1 calls with $n = 4$, $\lambda_1 = 0.2$, $\lambda_2 = 0.2$, $\mu_2 = 1.0$, $\alpha_1 = 1.0$, $\alpha_2 = 5.0$, $\omega = 10.0$, $c = 6$.

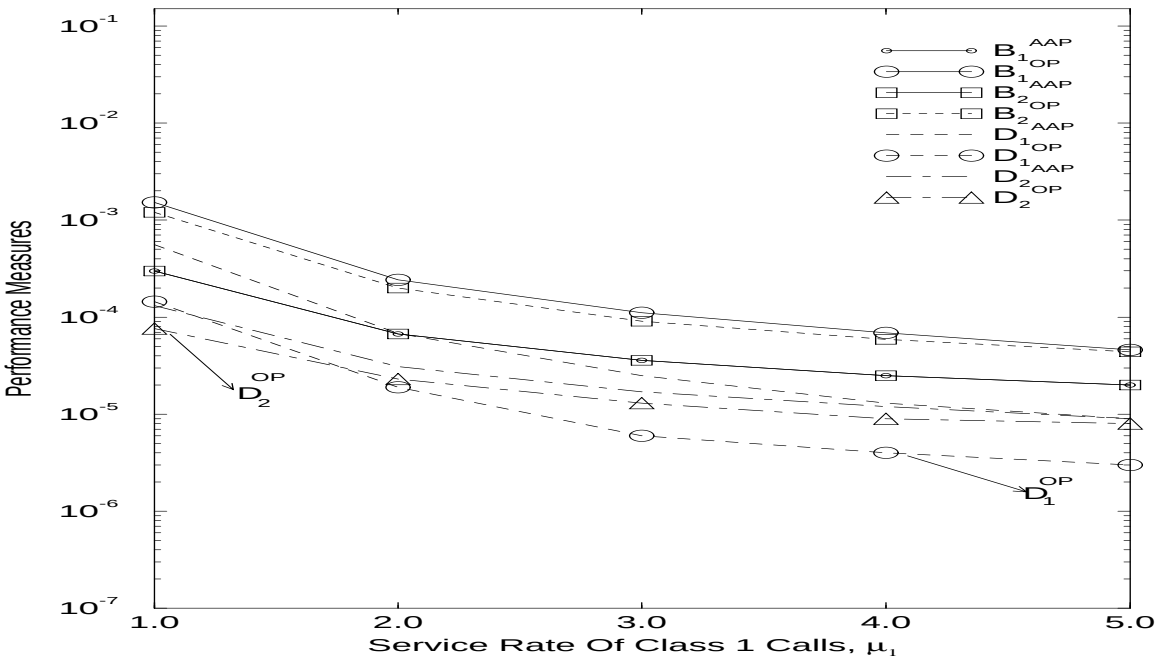

Figure 5.20: Performance evaluation of a two-class ring system by optimizing weighted call incompletion rate under variable service rate of class 1 calls with $n = 4$, $\lambda_1 = 0.1$, $\lambda_2 = 0.2$, $\mu_2 = 1.0$, $\alpha_1 = 5.0$, $\alpha_2 = 1.0$, $\omega = 10.0$, $c = 6$.

Chapter 6

Conclusion and Future Work

In this work, the MDP is applied to measure a variety of performance measures which show when optimal policies are needed to improve performance substantially and when not. We have evaluated performance measures under two system objectives – maximizing call completion rate and minimizing weighted call incompletion rate.

6.1 Summary of Results

A synopsis of our result follows:

- Maximization of the call completion rate did not provide any improvement of system performance for the single class system and this was also observed for the two-class ring system for any reasonable set of system parameters. For a small linear system (without handoffs) at higher load there was a small improvement in the optimal performance. This was due to the non-symmetric structure of the cells at the two ends. These effects on the overall system performance measures diminish as the size of the system increases. The policy of maximizing call completion rate never differs that much from the admit all policy. As a result high probability of handoff dropping was observed at higher speed and at higher load. We have considered low probability of handoff dropping as reliability of service. Therefore we have minimized the weighted call incompletion rate, where the weight implies the relative penalty for dropping a handoff over blocking a new call. The value of the weight was always greater than 1.

- In weighted call incompletion rate optimization problem, increasing the value of relative penalty reduces handoff dropping by blocking a higher number of new

calls, which was expected. For the two-class system it was observed that the class with higher mobility faces higher probability of blocking to enter the system but the probability of handoff dropping of both types depend on their relative arrival rate and relative service rate.

6.2 Suggestions for Further Study

This work does not include revenue maximization. But it can be easily done by defining suitable reward function for a single class system and as well as a two class system. For revenue maximization, like maximization of call completion rate, the optimal policy may not provide significant improvement in the system performance. Because in this situation no preferential attention will paid to either handoffs nor new calls. It was seen that under that restriction, the system can not follow a policy which is very different from the admit all policy. So, for revenue maximization high probability of handoff dropping may be observed at a high load and at a high speed. If this undesirable situation occurs, then introducing weight in the reward rate will be needed. We feel this area is particularly intersting from a business point of view and should be explored in some depth.

The MDP suffers from the curse of dimensionality. Indeed the number of states grows almost exponentially with the number of classes. Therefore the MDP algorithm becomes intractable for a cellular network which supports large number of classes of customers and has large number of cells. In this connection, we can explore the idea to investigate heuristics that locate high-performing suboptimal policies for this kind of system. This can be done by decomposing the bigger system into a number of smaller systems and finding the optimal policy locally. Finally the aggregation of all those local optimal policies may result into high-performing suboptimal policies for the whole system. But in this kind of situation the performance of the suboptimal policies probably will depend on the number of decomposed smaller systems. We leave this idea as future work.

References

[And73] L. G. Anderson. A Simulation Study of Some Dynamic Channel Assignment Algorithms in a High Capacity Mobile Telecommunications System. *IEEE Trans. Vech. Technol.*, 22:210–217, 1973.

[Bat73] J. Bather. Optimal Decisions Procedures for Finite Markov Chains. *Adv. Appl. Prob.*, 5:521–540, 1973.

[Bel57] R. Bellman. *Dynamic Programming.* Princeton University Press, Princeton, 1957.

[CFI] L. J. Cimini, G. F. Foschini, and Chih-Lin I. Call Blocking Performance of Distributed Algorithm for Dynamic Channel Allocation in Microcells.

[CR72] D. C. Cox and D. O. Reudink. Dynamic Channel Assignment in Two Dimensional Large Scale Mobile Radio System. *Bell Syst. Tech. J.*, 21:1611–1627, 1972.

[EM83] D. E. Everitt and N. W. Macfadyen. Analysis of Multicellular Mobile Radiotelephone System with Loss. *Britt. Telecommun. Technol. J.*, 1(2):37–45, Oct. 1983.

[EM89] D. E. Everitt and D. Manfield. Performance Analysis of Cellular Mobile Communication Systems with Dynamic Channel Assignment. *IEE Select. Areas Commun,*, 7:1172–1180, Oct. 1989.

[ESG82] S. M. Elnoubi, R. Singh, and S. C. Gupta. A New Frequency Channel Assignment Algorithm in High Capacity Mobile Communication Systems . *IEEE Trans. Veh. Technol.*, 31:125–131, 1982.

[FG83] G. J. Foschini and B. Gopinath. Sharing memory optimally. *IEEE Trans. Commun.*, 31(3):352–360, 1983.

[GS83] I. S. Gopal and T. E. Stern. Optimal call blocking policies in an integrated services environment. In *Conference on Information Sciences Systems*, pages 383–388. Johns Hopkins University, 1983.

[How60] R. A. Howard. *Dynamic Programming and Markov Processes.* M.I.T. Press, 1960.

[Kau81] J. S. Kaufman. Blocking in a Shared Resource Environment. *IEEE Trans. Commun.*, 29(10):1474–1481, 1981.

[KG78] T. J. Kahwa and N. D. Georganas. A Hybrid Channel Assignment Scheme in Large Scale Cellular-Structured Mobile Communication Systems. *IEEE Trans. Commun*, 26:432–438, 1978.

[Ray] P-A. Raymond. Performance Analysis of Cellular Networks. INRS Telecommunications (June 1990 preprint).

[Ros83] S. M. Ross. *Stochastic Processes*. John Wiley & Sons, New York, 1983.

[RT89a] K. W. Ross and D. Tsang. Optimal circuit access policies in an isdn environment: A markov decision approach. *IEEE Trans. Commun.*, 37(9):934–939, 1989.

[RT89b] K. W. Ross and D. Tsang. The stochastic knapsack problem. *IEEE Trans. Commun.*, 37(7):740–747, 1989.

[RY93] C. Rose and R.D. Yates. Optimal Call Admission to Single Cells of a Cellular Mobile Network. *IEEE Trans. Commun*, 1993. (submitted October 1,1993).

[SF79] P.J. Schweitzer and A. Federgruen. Geometric Convergence of Value-Iteration in Multichain Markov Decision Problems. *Adv. Appl. Prob.*, 11:188–217, 1979.

[SKK80] M. Sengoku, M. Kurata, and Y. Kajitani. Rearrangements of Switching Networks and its Application to a Mobile Radio Communication System. *IEEE Int. Conf. Circuits Comput.*, pages 402–405, 1980.

[Tij86] H. C. Tijms. *Stochastic Modelling and Analysis: a computational approach*. Wiley, New York, 1986.

[Yen94] A. Yener. *Genetic Algorithms and Call Admission in Cellular/Wireless Systems*. M.S. Thesis, Rutgers University, 1994.

[Zij86] W. H. M. Zijm. Asymptotic Expansions for Dynamic Programming in Recursions with General Nonnegative Matrices. *Optim. Theory and Appl.*, 48, 1986.